Fields Institute Monographs 33

The Fields Institute for Research in Mathematical Sciences

Thomas Lam · Luc Lapointe
Jennifer Morse · Anne Schilling
Mark Shimozono · Mike Zabrocki

k-Schur Functions and Affine Schubert Calculus

FIELDS

Springer

Fields Institute Monographs

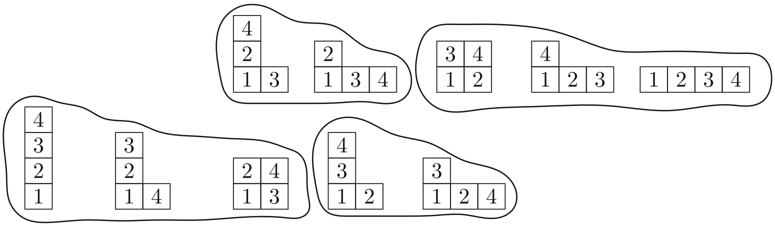
Volume 33

The Fields Institute is a centre for research in the mathematical sciences, located in Toronto, Canada. The Institutes mission is to advance global mathematical activity in the areas of research, education and innovation. The Fields Institute is supported by the Ontario Ministry of Training, Colleges and Universities, the Natural Sciences and Engineering Research Council of Canada, and seven Principal Sponsoring Universities in Ontario (Carleton, McMaster, Ottawa, Toronto, Waterloo, Western and York), as well as by a growing list of Affiliate Universities in Canada, the U.S. and Europe, and several commercial and industrial partners.

For further volumes:
http://www.springer.com/series/10502

Thomas Lam • Luc Lapointe • Jennifer Morse
Anne Schilling • Mark Shimozono
Mike Zabrocki

k-Schur Functions and Affine Schubert Calculus



FIELDS The Fields Institute for Research
in the Mathematical Sciences

Springer

Thomas Lam
Department of Mathematics
University of Michigan
Ann Arbor, MI, USA

Luc Lapointe
Instituto de Matemática y Física
Universidad de Talca
Talca, Chile

Jennifer Morse
Department of Mathematics
Drexel University
Philadelphia, PA, USA

Anne Schilling
Department of Mathematics
University of California
Davis, CA, USA

Mark Shimozono
Department of Mathematics
Virginia Tech
Blacksburg, VA, USA

Mike Zabrocki
Mathematics and Statistics
York University
Toronto, ON, Canada

ISSN 1069-5273 ISSN 2194-3079 (electronic)
ISBN 978-1-4939-0681-9 ISBN 978-1-4939-0682-6 (eBook)
DOI 10.1007/978-1-4939-0682-6
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2014934668

Mathematics Subject Classification (2010): 05E05, 05E10, 14R99, 14N15, 14N35

Cover illustration: Drawing of J.C. Fields by Keith Yeomans

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Chapter 1
Introduction

Affine Schubert calculus is a subject that lies at the crossroads of combinatorics, geometry, and representation theory. Its modern development is motivated by two seemingly unrelated directions. One is the introduction of k-Schur functions in the study of Macdonald polynomial positivity, a mostly combinatorial branch of symmetric function theory. The other direction is the study of the Schubert bases of the (co)homology of the affine Grassmannian, an algebro-topological formulation of a problem in enumerative geometry.

Classical Schubert calculus is a branch of enumerative algebraic geometry concerned with problems of the form:

How many lines L in 3-space intersect four fixed lines L_1, L_2, L_3, L_4?

In general, lines are replaced by affine linear subspaces, and conditions on the dimensions of intersections are imposed. When L_1, L_2, L_3, L_4 are in generic position, the answer to the above problem is two; this is a pleasant surprise, since in linear algebra one expects to find 0, 1, or ∞ solutions. Schubert [144] studied such "Schubert problems" in the nineteenth century. At the turn of the twentieth century, Hilbert posed as his 15th problem the rigorous foundation of Schubert's enumerative calculus. Subsequent developments in geometry and topology converted such Schubert problems into problems of computation in the cohomology ring $H^*(\mathrm{Gr}(k,n))$ of the Grassmannian $\mathrm{Gr}(k,n)$ of k-planes in n-space. The problems were reduced to finding structure constants, now called *Littlewood-Richardson coefficients* [118], of a certain "Schubert basis" for $H^*(\mathrm{Gr}(k,n))$.

The explicit realization of these computations using the theory of Schur functions played an important role in transforming Schubert calculus into a contemporary theory that stretches into many fields. The Schur functions s_λ form a basis for the symmetric function space Λ and at the turn of the century, it was discovered that they match irreducible representations of the symmetric group. Later, a deep connection between Schur functions and the geometry of Grassmannians was established when it was shown that the Schubert structure constants exactly equal coefficients in the product of Schur functions in Λ. The rich combinatorial backbone of the theory

T. Lam et al., *k-Schur Functions and Affine Schubert Calculus*, Fields Institute Monographs 33, DOI 10.1007/978-1-4939-0682-6_1,

of Schur functions, including the Robinson–Schensted algorithm, jeu-de-taquin, the plactic monoid (see for example [139]), crystal bases [127], and puzzles [74], now underlies Schubert calculus and in particular produces a direct formula for the Littlewood–Richardson coefficients. The influence of Schur functions on the geometry of Grassmannians provoked the broadening of Schubert calculus to other studies ranging from representation theory to physics.

A trend in Schubert calculus is to generalize the classical setup in two basic directions: (1) to vary the underlying geometric object being considered by replacing the Grassmannian by the flag variety, or more generally by a partial flag variety of a Kac–Moody group, and (2) to vary the algebraic structure considered by replacing cohomology by equivariant cohomology, K-theory, quantum cohomology, or other algebraic invariants. Our interest is in the case when the Grassmannian is replaced by infinite-dimensional spaces Gr_G known as affine Grassmannians.

Investigations of the quantum cohomology rings of flag varieties led Peterson [130] to begin a systematic study in this direction for any complex simple simply-connected algebraic group G. Applying work of Kostant and Kumar [75] on the topology of Kac-Moody flag varieties, Peterson showed that the equivariant homology $H_T(\mathrm{Gr}_G)$ is isomorphic to a subalgebra of Kostant and Kumar's nilHecke ring. Moreover, he proved that the Littlewood–Richardson coefficients of $H_T(\mathrm{Gr}_G)$ could be identified with the 3-point Gromov–Witten invariants of the flag variety of G. A classical result of Quillen [135] establishes that the affine Grassmannian Gr_G is itself homotopy-equivalent to the group ΩK of based loops into the maximal compact subgroup $K \subset G$. This places Gr_G in a unique position amongst the homogeneous spaces of all Kac–Moody groups. It endows $H_T(\mathrm{Gr}_G)$ with the structure of a Hopf algebra, and is also partly responsible for the important position that the affine Grassmannian has in geometric representation theory.

The aim of this book is to present ongoing work developing a theory of *affine Schubert calculus* in the spirit of classical Schubert calculus; here the Grassmannian is replaced by the affine Grassmannian. As with Schubert calculus, topics under the umbrella of affine Schubert calculus are vast, but now it is the combinatorics of a family of polynomials called k-Schur functions that underpins the theory.

The theory of k-Schur functions originated in the apparently unrelated study of *Macdonald polynomials*. Macdonald polynomials are symmetric functions over $\mathbb{Q}(q,t)$ that possess remarkable properties; the proofs of which have inspired deep work in many areas (e.g. double affine Hecke algebras [32], quantum relativistic systems [138], Hilbert schemes of points in the plane [61]). Macdonald conjectured in the late 1980s that the coefficients expressing Macdonald polynomials in terms of the Schur basis lie in $\mathbb{N}[q,t]$. Since then, the Macdonald/Schur transition coefficients have been intensely studied from a combinatorial, representation theoretic, and algebro-geometric perspective.

In one such study [91], Lapointe, Lascoux, and Morse found computational evidence for a family of new bases for subspaces Λ_k^t in a filtration $\Lambda_1^t \subset \Lambda_2^t \subset \cdots \subset \Lambda_\infty^t$ of Λ. Conjecturally, the star feature of each basis was the property that Macdonald polynomials expand positively in terms of it, giving a remarkable factorization for the Macdonald/Schur transition matrices over $\mathbb{N}[q,t]$. Pursuant investigations of these bases led to various conjecturally equivalent characterizations and the discovery that they refined the very aspects of Schur functions that make them so fundamental and wide-reaching. As such, they are now generically called *k-Schur functions*.

The role of k-Schur functions in affine Schubert calculus emerged over a number of years. The springboard was a realization that the combinatorial backbone of k-Schur theory lies in the setting of the type-A affine Weyl group. Generalizing the classical theory of Schur functions, Pieri rules, Young's lattice, the Cauchy identity, tableaux, and Stanley symmetric functions were refined using k-Schur functions [79, 92, 96]. These are naturally described in terms of posets of elements in $\tilde{A}_k$. For example, the number of monomial terms in an entry of the Macdonald/k-Schur matrix equals the number of reduced expressions for an element in $\tilde{A}_k$.

The combinatorial exploration fused into a geometric one when the k-Schur functions were connected to the quantum cohomology of Grassmannians. Lapointe and Morse [93] showed that each Gromov–Witten invariant for the quantum cohomology of Grassmannians exactly equals a k-Schur coefficient in the product of k-Schur functions in Λ. A basis of *dual (or affine) k-Schur functions* was also introduced in [93]. In response to questions about the geometric role for dual k-Schur functions and the significance of the complete set of k-Schur coefficients, Morse and Shimozono conjectured that the Schubert bases for cohomology and homology of the affine Grassmannian Gr are given by the dual k-Schur functions and the k-Schur functions, respectively. Lam proved the conjectures in [80]. Since then, the synthesis of affine Schubert calculus and k-Schur function theory has produced a subject involving prolific research in mathematics, computer science, and physics.

This book arose from an NSF funded Focused Research Group entitled "Affine Schubert Calculus: Combinatorial, geometric, physical, and computational aspects", which involved Thomas Lam, Luc Lapointe, Jennifer Morse, Anne Schilling, Mark Shimozono, Nicolas M. Thiéry, and Mike Zabrocki as active participants among others. Our exposition here grew out of several lecture series given at a summer school on 'Affine Schubert Calculus' organized by Anne Schilling and Mike Zabrocki and held in July 2010 at the Fields Institute in Toronto.

We give the story in three parts, through varying lenses. Chapter 2 presents the origins and early work on k-Schur functions, emphasizing the symmetric function setting and the combinatorics therein. The computational aspects are highlighted and illustrated with examples in SAGE [140, 151]. More information about the open-source computer algebra system SAGE is given in Appendix. Chapter 3 is Thomas Lam's synopsis of his summer school lectures entitled "affine Stanley symmetric functions". This chapter explains the combinatorial connections between Stanley symmetric functions and k-Schur functions via the algebraic constructions of nilCoxeter and nilHecke rings. Some of the latter constructions are presented for arbitrary root systems. Chapter 4 is Mark Shimozono's synopsis of his lectures on "generalizations to other affine group types". This chapter presents the nilHecke ring in the very general Kac-Moody setting and develops some of the geometric connections. The general construction is then applied to the situation of the affine Grassmannian. Each chapter is self-contained and can in principle be read independently of the others.

Let us outline the contents of this book. As discussed, the origin of k-Schur functions was in a study of Macdonald polyomials where they are characterized as symmetric functions that depend on one additional parameter t. However, the bulk of our presentation lies in the $t = 1$ setting. Although the general case is needed for implications in representation and Macdonald theory, the proven combinatorial and geometric properties largely center around this special case of k-Schur functions.

Extensive computer experimentation led to many conjectured properties of the k-Schur functions. Most notable is the k-Pieri rule for k-Schur functions, allowing one to express the product of a k-Schur function with a homogeneous symmetric function in terms of k-Schur functions. Chapter 2 starts by laying the combinatorial foundation needed to describe the k-Pieri rule including partitions, cores, and the affine Weyl group of type-A. Then, for fixed k and for $t = 1$, the k-Schur functions are presented as the family of symmetric functions which satisfy this Pieri rule. These functions form a basis of a subalgebra of the ring of symmetric functions. The dual basis lies in a Hopf-dual algebra which may be realized as a quotient of the ring of symmetric functions. Chapter 2 studies the k-Schur functions and their duals as symmetric functions, including a detailed summary of the *weak and strong tableaux* for which the k-Schur functions and their duals are the generating functions. Section 6 of this chapter includes an account of the affine insertion algorithm of [81], which explains how the generating functions for strong tableaux (k-Schur functions) are known to be dual to the generating function for weak tableaux (dual k-Schur functions).

For arbitrary t, the k-Schur functions span a subspace of the ring of symmetric functions which is closed under the coproduct operation. It was in this setting that the k-Schur functions originally arose. They were first defined as a sum of the usual Schur functions over a combinatorially defined collection of tableaux known as a k-atom. Lapointe, Lascoux, and Morse [91] conjectured that the Macdonald symmetric functions expand positively in terms of k-Schur functions. An obvious difficulty with this approach is a missing algebraic connection that could be used to connect Macdonald symmetric functions with the combinatorics of k-atoms. A second definition of the k-Schur functions was given in terms of symmetric function operators and followed in subsequent research [94, 95]. Chapter 2, Sect. 3 discusses these definitions as well as several others that are conjecturally equivalent. Section 4 is used to give a list of mostly conjectural properties of k-Schur functions and an account of what is known (to date) about the status of these conjectures.

Throughout Chap. 2 we have included examples of computations with SAGE in order to demonstrate examples of the formulas, but also to show how to use the functions that have been written by developers and incorporated into SAGE. These examples will hopefully both inspire and encourage exploration so that readers can generate further data and make new conjectures about k-Schur functions and their duals. We recommend to the reader to work with an up to date copy of SAGE (at least version 5.13 or later) to ensure that all features used in this book have been incorporated.

Chapter 3 then goes into more depth about k-Schur functions in the setting of the nil-Coxeter algebra. In the early 1980s, Stanley [150] became interested in the enumeration of the reduced words in the symmetric group. This led him to define a family of symmetric functions $\{F_w \mid w \in S_n\}$ now known as *Stanley symmetric functions*. In [79], Lam showed that the dual k-Schur functions were a special case of the affine Stanley symmetric functions $\tilde{F}_w$, analogues of Stanley's symmetric functions for the affine symmetric group.

In earlier work of Fomin and Stanley [48], it was shown that some of the main properties of Stanley symmetric functions could be obtained systematically from the nilCoxeter algebra of the symmetric group. This algebra is the associated graded algebra of the group algebra $\mathbb{C}[S_n]$ with respect to the length filtration. The affine nilCoxeter algebra played the same role for affine symmetric functions, and this provided an algebraic tool to study k-Schur functions and their duals. This interplay between algebra, combinatorics and symmetric functions is the main theme of Chap. 3. The connection to the nilHecke ring of Kostant and Kumar [75] is also explained and parts of the theory is carried out in the case of an arbitrary Weyl group.

Chapter 4 puts the preceding chapters in a more geometric context, and begins with a careful development of Kostant and Kumar's nilHecke ring $\mathbb{A}$ [75]. The nilHecke ring can roughly be described as the smash product of the nilCoxeter algebra and a polynomial ring and it was introduced to study the torus equivariant cohomology of Kac–Moody partial flag varieties. This ring acts as divided difference operators on the equivariant cohomology.

Peterson [130] studied the equivariant homology $H_T(\mathrm{Gr}_G)$ of the affine Grassmannian Gr_G of the complex simple simply-connected algebraic group G as a Hopf algebra with the following idea: applying the homotopy equivalences $\mathrm{Gr} \simeq \Omega K$ and $\mathrm{Fl}_{\mathrm{af}} \simeq LK/T_{\mathbb{R}}$ the natural inclusion $\Omega K \hookrightarrow LK/T_{\mathbb{R}}$ gives rise to an action of $H_T(\mathrm{Gr}_G)$ on $H_T(\mathrm{Fl}_{\mathrm{af}})$. (Here $\mathrm{Fl}_{\mathrm{af}}$ denotes the affine flag variety of G.) This action can be described in terms of divided difference operators, giving an injection $j : H_T(\mathrm{Gr}_G) \to \mathbb{A}$. Peterson's work is given a thorough treatment in Chap. 4, Sect. 4.

Using the natural relation between the nilCoxeter algebra and the nilHecke ring, in [79] Lam confirmed a conjecture of Morse and Shimozono identifying polynomial representatives for the Schubert classes of the affine Grassmannian as the k-Schur functions in homology and the dual k-Schur functions in cohomology. The algebraic part of this result is established in Chap. 3, Theorems 8.9 and 8.11.

We now discuss various generalizations of (dual) k-Schur functions, which are symmetric-function versions of the Schubert bases of the dual Hopf algebras of the homology $H_*(\mathrm{Gr}_{SL_n})$ and cohomology $H^*(\mathrm{Gr}_{SL_n})$ of the type A affine Grassmannian Gr_{SL_n}. For the affine Grassmannians Gr_G for G of classical type, analogous symmetric functions have been defined [85, 131]. However, only for the analogues of dual k-Schur functions, is an explicit monomial expansion known. The classical type analogues of k-Schur functions are only defined by duality and little is known about their combinatorics.

There is an equivariant or "double" analogue of k-Schur functions, called k-double Schur functions [89], which are to k-Schur functions what double Schubert polynomials are to Schubert polynomials. These are symmetric functions for the Schubert bases of the equivariant homology $H_T(\mathrm{Gr}_{SL_n})$ and $H^T(\mathrm{Gr}_{SL_n})$ for the "small torus" T, a maximal torus in G (as opposed to the maximal torus in the affine Kac-Moody group). The k-Schur functions are recovered from their double analogues by setting some variables to zero. Aside from setting up the correct symmetric function rings and bases, the only combinatorial result in this context is a Pieri rule for $H_T(\mathrm{Gr}_{SL_n})$.

Essentially all of the general theory presented here has an analogue in K-theory, which carries more information than (co)homology. Passing from the k-Schur function to its K-theoretic analogue, is like passing from a Schubert polynomial to a Grothendieck polynomial. For the affine Grassmannian, as in (co)homology one again has a pair of dual Hopf algebras, but one obtains two *pairs* of dual bases; both algebras have a structure sheaf basis and an ideal sheaf basis. Kostant and Kumar developed the torus-equivariant K-theory of Kac-Moody homogeneous spaces [76] and Peterson's theory can be carried out in K-theory as well [84]. In particular Peterson's j-basis (see Chap. 4, Sect. 4.5), which is defined algebraically using a leading term condition for an expansion in the divided difference basis, has an analogue (called the k-basis in [84]) that corresponds to ideal sheaves of Schubert varieties in the affine Grassmannian. Peterson's "quantum equals affine" theorem [90, 130] (see Chap. 4, Sect. 4.7) has an analogue in K-theory: the structure sheaves of opposite Schubert varieties in the quantum K-theory $QK^T(G/B)$ of finite-dimensional flag varieties G/B, appear to multiply in the same way as the structure sheaves of the Schubert varieties in the K-homology $K_T(\mathrm{Gr}_G)$ of the affine Grassmannian [83]. To establish this connection one must prove a conjectural Chevalley formula of Lenart and Postnikov [114, Conjecture 17.1] for quantum K-theory.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No. DMS-0652641. "Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation." We are grateful to the Fields Institute in Toronto for helping organize and support the summer school and workshop on "Affine Schubert calculus".

We would like to thank Tom Denton and Karola Mészáros for helpful comments and additions on Chap. 2, and Jason Bandlow, Chris Berg, Nicolas M. Thiéry as well as many other SAGE developers for their help with the SAGE implementations.

Appendix: SAGE

SAGE [151] is a completely open source general purpose mathematical software system, which appeared under the leadership of William Stein (University of Washington) and has developed explosively within the last 5 years. It is similar to MAPLE, MUPAD, MATHEMATICA, MAGMA, and up to some point MATLAB, and is based on the popular Python programming language. SAGE has gained strong momentum in the mathematics community far beyond its initial focus in number theory, in particular in the field of combinatorics, see [140].

Tutorials and instructions on how to install SAGE can be found at the main SAGE website http://www.sagemath.org/. For example, for the basic SAGE syntax and programming tricks see http://www.sagemath.org/doc/tutorial/programming.html.

Many aspects related to k-Schur functions and symmetric functions in general have been implemented in SAGE and in fact are still being developed as an on-going project. Throughout the text we provide many examples on how to use SAGE to do calculations related to k-Schur functions. Further information about the latest code and developments can be obtained from the SAGE-COMBINAT website [140]. We suggest that the interested reader uses SAGE version 5.13 or later to ensure that all features used in this book have been incorporated.

Chapter 2
Primer on k-Schur Functions

JENNIFER MORSE,[1] ANNE SCHILLING[2] AND MIKE ZABROCKI[3]
morsej@math.drexel.edu, anne@math.ucdavis.edu, zabrocki@mathstat.yorku.ca

based on lectures by Luc Lapointe and Jennifer Morse
lapointe@inst-mat.utalca.cl and morsej@math.drexel.edu

The purpose of this chapter is to outline some of the results and open problems related to k-Schur functions, mostly in the setting of symmetric function theory. This chapter roughly follows the outline of several talks given by Luc Lapointe and Jennifer Morse at a conference titled "Affine Schubert Calculus" held in July of 2010 at the Fields Institute in Toronto.[4]

In addition it presents many examples based on code written in SAGE [140, 151] by Jason Bandlow, Nicolas M. Thiéry, the last two authors, and many other SAGE developers. The following presentation is intended to give both an idea of the origins of the k-Schur functions as well as the current ideas and computational tools which have been most productive for demonstrating their properties.

We will present almost no proofs in this chapter, but rather refer to the original articles for detailed arguments. Instead the concepts are illustrated with many SAGE examples to highlight how to discover and experiment with many of the still open conjectures related to k-Schur functions. The purpose behind most of the SAGE examples is to demonstrate the formulas with examples and to give the commands that would allow a first time user of SAGE to be able to use the functions to generate data that they might need for their own research.

[1]The author was supported by NSF gant DMS–0652641, DMS–0652668, DMS–1001898.

[2]The author was supported by NSF grants DMS–0652641, DMS–0652652, DMS–1001256, and OCI–1147247.

[3]The author was supported by NSERC.

[4]see http://www.fields.utoronto.ca/programs/scientific/10-11/schubert/

T. Lam et al., *k-Schur Functions and Affine Schubert Calculus*, Fields Institute Monographs 33, DOI 10.1007/978-1-4939-0682-6_2,

Section 1 reviews much of the combinatorial background of k-Schur theory including partitions, cores, (partial) orders on the affine symmetric group, and some symmetric function theory. This section also sets up the combinatorial backdrop needed to give the Pieri rules for k-Schur functions and their duals. In Sect. 2, we define a parameterless ($t = 1$) family of k-Schur functions using an analogue of the Pieri rule for Schur functions [96]. This definition is used to relate k-Schur functions to the geometry and to Stanley symmetric functions discussed in Chap. 3. We also give the dual Pieri rule [81] which gives rise to a monomial expansion of the k-Schur functions. The Pieri and dual Pieri rule motivate the definition of weak and strong order tableaux.

In Sect. 3, we present four conjecturally equivalent definitions of the k-Schur functions for generic t. Some are known to be equivalent when $t = 1$. The first definition of k-Schur functions appeared in a paper by Lapointe, Lascoux and Morse [91] and is purely combinatorial in nature; defined as a sum over certain classes of tableaux called atoms. Lapointe and Morse [94] followed this paper by defining symmetric functions which were defined by algebraic operations instead of a sum over combinatorial objects. The last two definitions of the k-Schur functions with a generic parameter t are defined along lines similar to the parameterless k-Schur functions, but now a t-statistic is introduced on weak (resp. strong) order tableaux.

In Sect. 4 we present many of the properties of k-Schur functions and outline what is known about which property for each of the definitions. This is followed by Sect. 5 which contains further research directions and many conjectures that remain to be resolved (and hence the content is likely to change in the future)! Section 6 explains the duality between strong and weak order in terms of a k-analogue of the Robinson–Schensted–Knuth algorithm, which gives rise to an affine insertion algorithm. We present part of this algorithm by giving a bijection between permutations and pairs of tableaux. Finally in Sect. 7 some details about the branching from k to $(k + 1)$-Schur functions are given.

1 Background and Notation

1.1 Partitions and Cores

A *partition* $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{\ell(\lambda)})$ of m is a sequence of weakly decreasing positive integers which sum to $m = \lambda_1 + \lambda_2 + \cdots + \lambda_{\ell(\lambda)}$. The value of m is called the *size* of the partition and this will be denoted by $|\lambda|$. The entries of the partition are called the parts and the number of parts of the partition is denoted by $\ell(\lambda)$. As a general convention, if $i > \ell(\lambda)$ then $\lambda_i = 0$ and the definition of symmetric functions (which turn out to be indexed by partitions) given later in this section respects this convention. The statistic $n(\lambda) = \sum_{i=1}^{\ell(\lambda)} (i-1)\lambda_i$ on partitions has a value between 0 and $m(m-1)/2$ for partitions of m and this will arise in the definitions of symmetric functions. A partition λ is called *k-bounded* if $\lambda_1 \leq k$. The notation $\lambda \vdash m$ indicates that λ is a partition of m and generally we reserve the symbols λ, μ, ν to denote partitions.

A partition will be identified with its *Young (or Ferrers) diagram*. This is a diagram consisting of square cells arranged in left justified rows stacked on top of each other with the largest row with λ_1 cells on the bottom. (This convention is also called the French notation; when stacking the rows with the largest row at the top is called the English convention). Alternatively, a Young diagram is a collection of cells in the first quadrant of the (x, y)-plane with $\mathrm{dg}(\lambda) = \{(i, j) : 1 \leq i \leq \ell(\lambda) \text{ and } 1 \leq j \leq \lambda_i\}$ represented as boxes in the Cartesian plane so that the upper right hand corner of a cell has coordinate which is in this collection. For consistency with other references we have chosen that the first coordinate represents the row and the second coordinate represents the column (each beginning at 1 for the first row and column). For an example the Young diagram for the partition $\lambda = (4, 3, 3, 3, 2, 2, 1)$ is drawn in Example 1.1.

There is a partial order on partitions that arises naturally in symmetric functions when ordering basis elements. For two partitions λ, μ such that $|\lambda| = |\mu|$, we say that $\lambda \leq \mu$ if $\sum_{i=1}^{r} \lambda_i \leq \sum_{i=1}^{r} \mu_i$ for all $r \geq 1$. This is usually referred to as the *dominance order* on partitions.

The *conjugate* of a partition λ is the sequence $\lambda' = (\lambda'_1, \lambda'_2, \ldots, \lambda'_{\lambda_1})$ where $\lambda'_r = \#\{i : \lambda_i \geq r\}$. Alternatively, this can be seen on Young diagrams by reflecting the diagram in the $x = y$ line of the coordinate plane so that $\mathrm{dg}(\lambda') = \{(j, i) : (i, j) \in \mathrm{dg}(\lambda)\}$. For example in Example 1.1 below, $\lambda' = (7, 6, 4, 1)$ for the partition $\lambda = (4, 3, 3, 3, 2, 2, 1)$.

For many uses we will need to refer to the number of parts of a partition of a given size i and this will be denoted by $m_i(\lambda) = \#\{j : \lambda_j = i\}$. The quantity

$$z_\lambda = \prod_{i \geq 1} m_i(\lambda)!\ i^{m_i(\lambda)} \tag{1.1}$$

is the size of the stabilizer of a permutation $\sigma \in S_m$, the symmetric group on $m = |\lambda|$ letters, whose cycle type is λ under the conjugation action of S_m. That is, if σ has cycle type λ, then $z_\lambda = \#\{\tau \in S_m : \tau\sigma\tau^{-1} = \sigma\}$. Since we know that all permutations with the same cycle type are conjugate, the number of permutations with cycle type λ is equal to $m!/z_\lambda$.

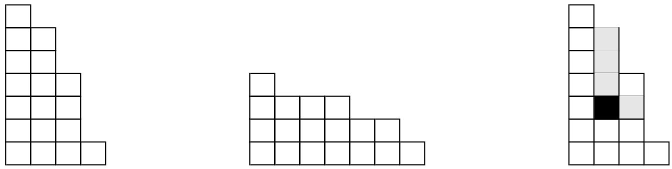
Each cell in a partition λ has a *hook length* which consists of the number of cells in the column above and in the row to the right (including the cell itself). Namely, for a cell $(i, j) \in \mathrm{dg}(\lambda)$, the hook length of the cell is $\mathrm{hook}_\lambda(i, j) = \lambda_i + \lambda'_j - i - j + 1$. In Example 1.1 below $\mathrm{hook}_{(4,3,3,3,2,2,1)}(3, 2) = 5 = \lambda_3 + \lambda'_2 - 3 - 2 + 1$.

For a partition λ with $\lambda_1 \leq k$, define the *k-split* of λ as a sequence of partitions (which will be denoted by $\lambda^{\to k}$) recursively. If $\lambda_1 + \ell(\lambda) - 1 \leq k$, then $\lambda^{\to k} = (\lambda)$. Otherwise,

$$\lambda^{\to k} = ((\lambda_1, \lambda_2, \ldots, \lambda_{k-\lambda_1+1}), (\lambda_{k-\lambda_1+2}, \lambda_{k-\lambda_1+3}, \ldots, \lambda_{\ell(\lambda)})^{\to k})\,. \tag{1.2}$$

In other words, the k-split of a partition is found by successively splitting off parts of the partition with hook k, starting with the first part, until that is no longer possible.

Example 1.1. The Young diagram for the partition $\lambda = (4, 3, 3, 3, 2, 2, 1)$ is the diagram on the left and its conjugate partition $\lambda' = (7, 6, 4, 1)$ is the diagram in the center.



The diagram on the right is the Young diagram for the partition $\lambda = (4, 3, 3, 3, 2, 2, 1)$ with the cells that are in the hook of the cell $(3, 2)$ shaded in. In this case $\mathrm{hook}_\lambda(3, 2) = 5$. The 4-split of λ is $\lambda^{\to 4} = ((4), (3, 3), (3, 2), (2, 1))$ and the 5-split is $\lambda^{\to 5} = ((4, 3), (3, 3, 2), (2, 1))$.

We will use the realization of the Young diagram as the set of cells in our notation and define $\lambda \subseteq \mu$ if $\mathrm{dg}(\lambda) \subseteq \mathrm{dg}(\mu)$. This forms a lattice, also known as the *Young lattice*, on set of partitions and the cover relation is given by $\lambda \to \mu$ if $\lambda \subseteq \mu$ and $|\lambda| + 1 = |\mu|$. The lattice is graded by the size of the partition and the first six levels of the infinite Hasse diagram are shown in Fig. 2.1.

There are several special types of containments of partitions that will arise in this discussion. If $\lambda \subseteq \mu$, then μ/λ is called a *skew partition* and it will represent the cells which are in $\mathrm{dg}(\mu)/\mathrm{dg}(\lambda)$, with the / here representing the difference of sets.

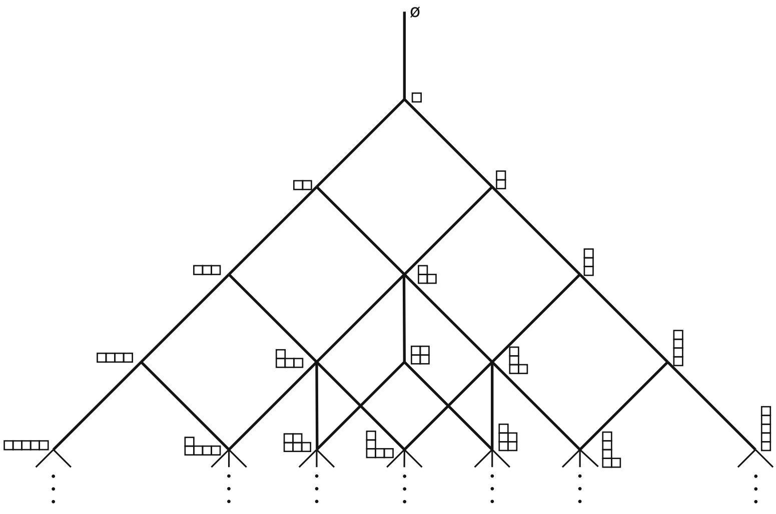


Fig. 2.1 The Young lattice of partitions (up to those of size 5) ordered by inclusion

We call μ/λ *connected* if for any two cells there is a sequence of cells in μ/λ from one to the other where consecutive cells share an edge. We say that μ/λ is a *horizontal* (*vertical*) strip if there is at most one cell in each column (row) of μ/λ. The skew partition μ/λ is called a *ribbon* if it does not contain any 2×2 subset of cells.

Sage Example 1.2. We now demonstrate how to access partitions and their properties in the open source computer algebra system SAGE (see section "Appendix: SAGE" in Chap. 1). We begin by listing all partitions of 4:

```
sage: P = Partitions(4); P
Partitions of the integer 4
sage: P.list()
[[4], [3, 1], [2, 2], [2, 1, 1], [1, 1, 1, 1]]
```

SAGE has list comprehension so that the last line could have also been written as

```
sage: [p for p in P]
[[4], [3, 1], [2, 2], [2, 1, 1], [1, 1, 1, 1]]
```

We can check how two partitions λ and μ relate in the dominance order

```
sage: la=Partition([2,2]); mu=Partition([3,1])
sage: mu.dominates(la)
True
```

and draw the entire Hasse diagram

```
sage: ord = lambda x,y: y.dominates(x)
sage: P = Poset([Partitions(6), ord], facade=True)
sage: H = P.hasse_diagram()
sage: view(H)        #optional
```

which outputs the graph. The `view(H)` command may not work properly unless dot2tex and Graphviz are installed on your version of Sage. Here we used the python syntax for a function, which is `lambda x : f(x)` for a function that maps x to $f(x)$. We can also compute the conjugate of a partition, its k-split

```
sage: la=Partition([4,3,3,3,2,2,1])
sage: la.conjugate()
[7, 6, 4, 1]
sage: la.k_split(4)
[[4], [3, 3], [3, 2], [2, 1]]
```

and create skew partitions

```
sage: p = SkewPartition([[2,1],[1]])
sage: p.is_connected()
False
```

1.2 Bounded Partitions, Cores, and Affine Grassmannian Elements

We will see that k-Schur functions are symmetric functions indexed by k-bounded partitions and consequently, the underlying combinatorial framework we need often comes out of a refinement of classical ideas in the theory of partitions. As it happens, the set of k-bounded partitions is in bijection with several different sets of natural combinatorial objects and often the k-Schur function setting is better expressed in those terms. To this end, we begin with a discussion of several other examples of possible indexing sets.

As with the k-bounded partitions, we are interested in another special subset of partitions. In particular, an *r-core* is a shape where none of its cells have a hook-length equal to r. We denote the set of all r-cores by $\mathcal{C}_r$. When we consider a partition as a core, the notion of size differs from the usual notion (where size counts the number of cells in the shape). In contrast, the relevant notion of size on a $(k+1)$-core is to count only the number of cells which have a hook-length smaller than $k+1$. We call this the *length* of the core. For a $(k+1)$-core κ, its length will be denoted by $|\kappa|_{k+1}$ or simply $|\kappa|$ if it is clear from the context that κ is viewed as a $(k+1)$-core. As $k \to \infty$, this becomes the usual size of the partition. Later in this section, we will see that the length is related to the length of elements in the affine symmetric group. Now, we give the connection between cores and bounded partitions.

Proposition 1.3 ([96, Theorem 7]). *There is a bijection between the set of $(k+1)$-cores κ with $|\kappa|_{k+1} = m$ and partitions $\lambda \vdash m$ with $\lambda_1 \leq k$.*

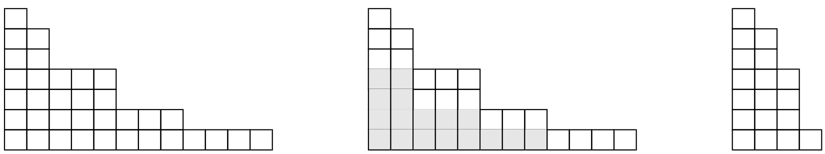
The bijection from $(k+1)$-cores to k-bounded partitions is

$$\mathfrak{p} : \kappa \mapsto \lambda\,,$$

defined by setting

$$\lambda_i = \#\{(i,j) \in \kappa : \mathrm{hook}_\kappa(i,j) \leq k\}\,. \tag{1.3}$$

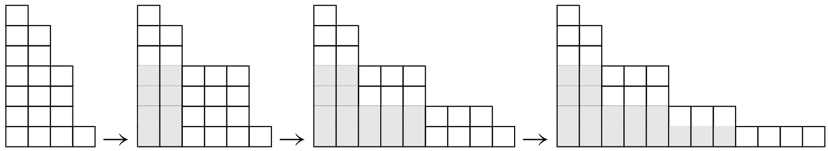
Example 1.4. The partition $(12, 8, 5, 5, 2, 2, 1)$ on the left is a 5-core since there are no cells in its Ferrers diagram with hook-length equal to 5. Equation (1.3) tells us how to applying $\mathfrak{p}$ to this core to obtain a 4-bounded partition; delete each cell in the diagram for the 5-core whose hook-length exceeds 5 and then slide all remaining cells to the left.



The first part of the resulting partition is at most 4.

The other direction of the bijection is also not difficult. Consider a k-bounded partition and work from the smallest part of the partition to the largest and slide the cells to the right until it is a $(k+1)$-core. Here is a description of the procedure which can be followed with Example 1.5. Start with the top row $\lambda_{\ell(\lambda)}$ of the k-bounded partition λ and successively move down a row. For a given row, calculate the hook lengths of its cells; if there is a cell with hook length greater than k, slide this row to the right until all cells have hook length less than or equal to k. Continue this process until all rows have been adjusted. The end result will be a $(k+1)$-core which we shall denote by $\mathfrak{c}_k(\lambda)$ or just $\mathfrak{c}(\lambda)$ if k is clear from the context.

Example 1.5. The partition $(4,3,3,3,2,2,1)$ is a 4-bounded partition. Here we draw the successive slides of the rows until we reach a 5-core:



Sage Example 1.6. Here is the way to compute the map $\mathfrak{c}$ in SAGE:

```
sage: la = Partition([4,3,3,3,2,2,1])
sage: kappa = la.k_skew(4); kappa
[12, 8, 5, 5, 2, 2, 1] / [8, 5, 2, 2]
```

For the inverse $\mathfrak{p}$ we write

```
sage: kappa.row_lengths()
[4, 3, 3, 3, 2, 2, 1]
```

If one is only handed the 5-core $(12,8,5,5,2,2,1)$ instead of the skew partition, one can do the following:

```
sage: tau = Core([12,8,5,5,2,2,1],5)
sage: mu = tau.to_bounded_partition(); mu
[4, 3, 3, 3, 2, 2, 1]
sage: mu.to_core(4)
[12, 8, 5, 5, 2, 2, 1]
```

All 3-cores of length 6 can be listed as:

```
sage: Cores(3,6).list()
[[6, 4, 2], [5, 3, 1, 1], [4, 2, 2, 1, 1], [3, 3, 2, 2, 1, 1]]
```

We now turn our attention to the third set of objects that is in bijection with the set of k-bounded partitions (and the set of $(k+1)$-cores). These come out of studying the type A affine Weyl group and its realization as the *affine symmetric group* $\tilde{S}_n$ given by generators $\{s_0, s_1, \ldots, s_{n-1}\}$ satisfying the relations