Parking Functions and Related Combinatorial Structures

by

Amarpreet Rattan

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Mathematics

in

Combinatorics and Optimization

Waterloo, Ontario, Canada, 2001

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The central topic of this thesis is parking functions. A parking function is a sequence of positive integers $(a_1, a_2, \ldots, a_n)$ such that its non-decreasing rearrangement $(b_1, b_2, \ldots, b_n)$ satisfies $b_i \leq i$. We give a survey of some of the current literature concerning these sequences and focus on their interaction with other combinatorial objects; namely noncrossing partitions, hyperplane arrangements and tree inversions. In the final chapter, we discuss generalizations of both parking functions and the above structures.

Acknowledgements

There are several people that I would like acknowledge for their help with this thesis. I would like to thank my fellow graduate students, in particular Simon Alexander, Jonathan Dumas, John Irving, James Muir and Antoine Vella. Their help, with both the mathematics and the technical writing of the thesis, was greatly appreciated. I would also like to thank my readers, Chris Godsil and David Wagner, for their time and helpful suggestions. Finally, I would like to give special thanks to my supervisor Ian Goulden whose constant availability, often at odd hours and short notice, and encouragement made the writing of this thesis a more rewarding and pleasant experience.

Contents

List of Figures

Chapter 1

Introduction

The central mathematical objects of this thesis are *parking functions*. A *parking function* is a sequence $(a_1, a_2, \ldots, a_n)$ of positive integers whose non-decreasing rearrangement $(b_1, b_2, \ldots, b_n)$ satisfies $b_i \leq i$. Parking functions, from their definition, seem to be simple sequences but they have arisen in, surprisingly, diverse areas of mathematics. They were first studied by Konheim and Weiss [9] in an occupancy problem in computer science but since have attracted attention as interesting objects in their own right.

It turns out that the number of parking functions of length n is $(n+1)^{n-1}$. One immediately recognizes this as the tree number of the complete graph on $n+1$ vertices or, equivalently, the number of trees on a fixed vertex set of cardinality $n+1$. There exists very simple bijections and other basic techniques showing that the number of parking functions of length n and the number of trees on $n+1$ vertices are the same, but we will be concentrating on how parking functions interact with a notion that is a refinement of trees, namely tree inversions. Parking functions and their interaction with tree inversions is the focus of Chapter 5.

The number $(n+1)^{n-1}$ also appears in the study of (at least) two other combinatorial structures and they are noncrossing partitions and hyperplane arrangements. It is known that the set of noncrossing partitions of $\{1, 2, \ldots, n+1\}$ is a subposet of the poset of partitions of $\{1, 2, \ldots, n+1\}$. It turns out that the number of maximal chains in the poset of noncrossing partitions is $(n+1)^{n-1}$. We show this by exhibiting a bijection with parking functions of the appropriate length. Further, associated with parking functions is a symmetric function PF_n which will be closely related to a symmetric function F_{NC_n} that is associated with noncrossing partitions. Noncrossing partitions are dealt with in Chapter 3. Concerning hyperplane arrangements, we will be talking about two hyperplane arrangements, the *braid arrangement* and the *Shi arrangement*. It is the Shi arrangement that has links with parking functions and our discussion of the braid arrangement is more or less needed to understand the Shi arrangement. In addition, a generating function associated with hyperplane arrangements, known as the *distance enumerator*, is shown to be connected with parking functions. Chapter 4 is devoted to hyperplane arrangements.

The final chapter, Chapter 6, deals with two generalizations of parking functions. We will show that there are generalizations of noncrossing partitions, hyperplane arrangements and tree inversions that fit with at least one of the two generalizations of parking functions given in Chapter 6.

Chapter 2

Fundamental Concepts

2.1 Background

In the following subsections we introduce some necessary terminology that will be used later. This section may be skipped by those who feel comfortable with the material. The notation is consistent with the following pieces of literature. The notation in Sections 2.1.2 and 2.1.1, on representation theory and symmetric functions, is consistent with Sagan [15] Macdonald [12] whereas the the notation in Section 2.1.3 on partially ordered sets (posets) is consistent with Stanley [20][21]. These sections are in no way meant to be complete, they merely introduce the material and the notation.

2.1.1 Symmetric Functions

A *partition* λ *of* n is a sequence of non-negative integers $(\lambda_1, \lambda_2, \ldots, \lambda_l)$ in non-increasing order that sum to n. Each positive λ_i is called a *part* of λ. The number of positive entries in λ, $l(\lambda)$, is called the *length* of λ. We use $\lambda \vdash n$ to mean that λ is a partition of n. Define $m_i(\lambda)$ to be the number of parts of λ equal to i and we will often write $\lambda = 1^{m_1(\lambda)}2^{m_2(\lambda)} \cdots n^{m_n(\lambda)}$, indicating that the number of i's occurring in λ is $m_i(\lambda)$. A sequence α of non-negative integers is said to have *shape* λ if its non-increasing rearrangement is λ and we use $\mathrm{sh}(\alpha)$ to mean the shape of α. Let $x = x_1, x_2, \ldots$ and for any sequence $\alpha = (\alpha_1, \alpha_2, \ldots)$, we denote by $\mathbf{x}^\alpha$ the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \cdots$. For the rest of this section, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \vdash n$.

For our purposes, a symmetric function $f(\mathbf{x})$ is a formal power series in a countable number of variables (which we assume to be $x_1, x_2, \ldots$) such that $(i\,j)f(\mathbf{x}) = f(\mathbf{x})$, where $(i\,j)f(\mathbf{x})$ is the series obtained by transposing the variables x_i and x_j in $f(\mathbf{x})$. The set of symmetric functions forms a ring which we call Λ. Furthermore, the ring of symmetric functions happens to be a vector space that has several different bases, which we now define.

The *monomial symmetric functions* are the symmetric functions, indexed by partitions γ of n,

$$m_\gamma = \sum_{\alpha\,:\,\mathrm{sh}(\alpha)=\gamma} \mathbf{x}^\alpha$$

The set $\{\, m_\gamma \mid \gamma \vdash n, n \geq 0\}$ of monomial symmetric functions forms a basis for Λ.

The *one part elementary symmetric functions*, *one part complete symmetric functions* and the *one part power sum symmetric functions* are the symmetric functions, indexed by non-negative integers,

$$e_r = \sum_{1 \leq i_1 < i_2 < \cdots < i_r} x_{i_1} x_{i_2} \ldots x_{i_r},$$

$$h_r = \sum_{1 \leq i_1 \leq i_2 \leq \cdots \leq i_r} x_{i_1} x_{i_2} \ldots x_{i_r}$$

and

$$p_r = \sum_{i \geq 1} x_i^r,$$

respectively, and we define e_0, h_0, and p_0 to equal 1. The sets $\{ e_r \mid r \geq 0\}$, $\{ h_r \mid r \geq 0\}$ and $\{ p_r \mid r \geq 0\}$ generate Λ. Further, we define

$$e_\lambda = e_{\lambda_1} e_{\lambda_2} \ldots e_{\lambda_n},$$

$$h_\lambda = h_{\lambda_1} h_{\lambda_2} \ldots h_{\lambda_n}$$

and

$$p_\lambda = p_{\lambda_1} p_{\lambda_2} \ldots p_{\lambda_n}$$

as the *elementary symmetric functions*, *complete symmetric functions* and *power sum symmetric functions*, respectively. The sets $\{ e_\lambda \mid \lambda \vdash n, n \geq 0\}$, $\{ h_\lambda \mid \lambda \vdash$

$n, n \geq 0\}$ and $\{ p_\lambda \mid \lambda \vdash n, n \geq 0\}$ are all bases for Λ.

The last type of symmetric functions that we will be using are the *Schur symmetric functions*. The Schur symmetric functions, indexed by a partition λ of n, for positive n, can be defined as

$$s_\lambda = \det(h_{\lambda_i - i + j})_{1 \leq i < j \leq n}$$

A function on Λ that will be used later is $\omega : \Lambda \longrightarrow \Lambda$ which maps e_r to h_r. It can be easily shown that ω is an algebra isomorphism. It turns out ω is its own inverse.

2.1.2 Group Representations and Permutation Groups

Let GL_d be the general linear group of dimension d (the set of all invertible $d \times d$ matrices) over the field $\mathbb{C}$. Given any group G, a *matrix representation of* G is a group homomorphism

$$X : G \longrightarrow GL_d,$$

or equivalently, X satisfies

1. $X(\epsilon) = I$, where ϵ is the identity in G and I is the identity matrix in GL_d.

2. $X(gh) = X(g)X(h)$ for all $g, h \in G$.

The parameter d is called the dimension of the representation. We may also write $GL(V)$ for GL_d, where V is a d dimensional vector space. Equivalently, we can use the language of modules to describe a representation. That is, a vector space V is a *G-module* if there is a multiplication $g\mathbf{v}$ of elements in V by elements of G such that

1. $g\mathbf{v} \in V$,
2. $g(c\mathbf{v} + d\mathbf{w}) = cg\mathbf{v} + dg\mathbf{w}$,
3. $(gh)\mathbf{v} = g(h\mathbf{v})$,
4. $e\mathbf{v} = \mathbf{v}$, where e is the identity of G.

for all $g, h \in G$, $\mathbf{v}, \mathbf{w} \in V$ and scalars c, d.

The following two representations will be used later.

Example 2.1 The simplest representation is the *trivial representation.* This the representation

$$X : G \longrightarrow GL_1$$

such that $X(g) = [1]$ for all $g \in G$. This is clearly a representation. $\square$

Example 2.2 The *permutation representation* is obtained when a group G acts on a set S. We take the vector space $\mathbb{C}[S] = c_1s_1+c_2s_2+\cdots+c_ns_n$ where $c_i \in \mathbb{C}$ and $S = \{s_1, s_2, \ldots, s_n\}$. Letting $v = c_1s_1 + c_2s_2 + \cdots + c_ns_n$ then $X(g)$ is defined as the matrix associated with the linear transformation $g.v = c_1g.s_1 + c_2g.s_2 + \ldots + c_ng.s_n$, where $g.s_i$ is g acting on s_i, with respect to the basis $(s_1, s_2, \ldots, s_n)$. $\square$

Another concept that we will be using is that of an induced representation. Suppose that H is a subgroup of the group G and that $t_1, t_2, \ldots, t_n$ is a *transversal* (i.e. the sets $t_1H, t_2H, \ldots, t_mH$ are all disjoint and $t_1H \cup t_2H \cup \ldots \cup t_mH = G$). Further suppose that $X : H \longrightarrow GL_d$ is a representation of the group H. Then the *induced representation* $\mathrm{ind}_H^G X : G \longrightarrow GL_{md}$ is defined as

$$\mathrm{ind}_H^G X(g) = \begin{bmatrix} X(t_1gt_1^{-1}) & X(t_1gt_2^{-1}) & \cdots & X(t_1gt_m^{-1}) \\ X(t_2gt_1^{-1}) & X(t_2gt_2^{-1}) & \cdots & X(t_2gt_m^{-1}) \\ \vdots & \vdots & \ddots & \vdots \\ X(t_mgt_1^{-1}) & X(t_mgt_2^{-1}) & \cdots & X(t_mgt_m^{-1}) \end{bmatrix}$$

where, in the matrix, $X(j)$ is the zero matrix if $j \notin H$.

The group that we will be interested in is the symmetric group, $\mathfrak{S}_n$. We will use either the standard cycle representation of a permutation (writing a permutation as the product of cycles) or writing a permutation as a word. We denote by $\mathfrak{S}_n$ the symmetric group on n symbols. If H is a subgroup of $\mathfrak{S}_n$ acting on a set S then the *orbit* of an element $s \in S$ is the set $\{ g.s \mid g \in H\}$. The action of H on S is said to be *transitive* if there is only one orbit, i.e. for any $s \in S$ the orbit of s is S. Often we wish to specify a set other than $\{1, 2, \ldots, n\}$ as our underlying set for the symmetric group. Thus, we write $\mathfrak{S}_A$ where A is a set with cardinality n to mean the symmetric group has as its underlying set the symbols in A. A useful subgroup of $\mathfrak{S}_n$ is the *Young Subgroup* $\mathfrak{S}_\lambda$, where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l) \vdash n$ which is the subgroup defined as

$$\mathfrak{S}_{\{1,2,\ldots,\lambda_1\}} \times \mathfrak{S}_{\{\lambda_1+1,\lambda_1+2,\ldots,\lambda_1+\lambda_2\}} \times \cdots \times \mathfrak{S}_{\{n-\lambda_l+1,n-\lambda_l+2,\ldots,n\}}.$$

2.1.3 Partially Ordered Sets

A *poset* P is an ordered pair $(P, \leq_P)$ where P is a set and $\leq$ is a reflexive, transitive and anti-symmetric relation on the set P. We note the abuse of notation of calling both the poset P and its underlying set P the same thing. This abuse, will, in fact turn out to be convenient. Given a poset P we call $Q = (Q, \leq_Q)$ a subposet of P if $Q \subseteq P$ and for $x, y \in Q$, $x \leq_Q y$ if and only if $x \leq_P y$. We say that x *covers* y if $x > y$ and no z satisfies $x > z > y$. For any $x, y \in P$ we denote by $[x, y]$ the subposet of P whose underlying set is $\{ z \mid x \leq z \leq y\}$. We call $[x, y]$ an *interval.* If P contains an x such that $x \leq y$ for all $y \in P$ we call x the $\hat{0}$ of P. Similarly, if P contains an x such that $x \geq y$ for all $y \in P$ then we denote x by $\hat{1}$. P is called a *chain* if any two elements of P are comparable. A *chain in* P is just a subposet of P that is a chain. If $\hat{0}, \hat{1} \in P$ and if every maximal chain in P has the the same length, say n, then we call P a *graded poset of rank* n. In that case, there exists a unique function $\rho : P \longrightarrow \mathbb{N}$ (which we will call the rank function) such that $\rho(\hat{0}) = 0$ and $\rho(x) = \rho(y) + 1$ whenever x covers y. If P is graded of rank n then P is called *rank symmetric* if the number of elements of rank i is the same as the number of elements of rank $n - i$. Further, P is *locally rank symmetric* if every interval is rank symmetric. The *dual* of a poset P, P^*, is the poset on the same set as P and $x \leq y$ in P^* if and only if $y \leq x$ in P. P is called *self-dual* if $P = P^*$.

Given two posets P and Q define the *direct product* of P and Q, $P \times Q$, on the set $\{ (x, y) \mid x \in P, y \in Q\}$ and $(x, y) \leq (x', y')$ if and only if $x \leq x'$ in P and $y \leq y'$ in Q.

We note some obvious facts. If P is graded and self-dual, it is rank symmetric. Further, the product of two self-dual posets P and Q is also self-dual and, hence, rank symmetric. We will be using this fact later.

Suppose that P is a graded poset of rank n with rank function ρ. Let $\rho(s, t)$ be shorthand for $\rho(t) - \rho(s)$ and define F_P to be the formal power series

$$F_P(x) = \sum_{\hat{0} \leq t_0 \leq t_1 \leq \ldots \leq t_{k-1} < t_k = \hat{1}} x_1^{\rho(t_0,t_1)} x_2^{\rho(t_1,t_2)} \ldots x_k^{\rho(t_{k-1},t_k)} \tag{2.1}$$

where the sum is over all multichains that contain $\hat{1}$ precisely once (this ensures that (2.1) is, in fact, a formal power series i.e. that each monomial has a finite coefficient). For the set $S = \{m_1, m_2, \ldots, m_j\}$ we write $S = \{m_1, m_2, \ldots, m_j\}_<$ to mean that $m_1 < m_2 < \ldots < m_j$. We define $\alpha_P(S)$ to be the number of chains $\hat{0} < t_1 < t_2 \ldots < t_j < \hat{1}$ such that $S = \{\rho(t_1), \rho(t_2), \ldots, \rho(t_j)\}$. Further, for any partition $\lambda \vdash n$, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ we define $S_\lambda = \{\lambda_1, \lambda_1 + \lambda_2, \ldots, \lambda_1 + \lambda_2 + \ldots + \lambda_{l-1}\}$. An immediate result of the above definition of $F_P(x)$ is the following.

Proposition 2.3

$$F_P(x) = \sum_{\substack{\{m_1, m_2, \ldots, m_j\}_< \\ S \subseteq [n-1]}} \sum_{1 \leq i_1 < i_2 < \ldots < i_{j+1}} x_{i_1}^{m_1} x_{i_2}^{m_2 - m_1} \ldots x_{i_{j+1}}^{n-m_j} \alpha_P(S)$$

and $F_P(x)$ is a symmetric function if and only if

$$F_P(x) = \sum_{\lambda \vdash n} \alpha_P(S_\lambda) m_\lambda.$$

2.2 Definitions

We begin with a definition of parking functions.

Definition 2.4 *A parking function of length n is a sequence $(a_1, a_2, \dots, a_n)$ of positive integers such that its non-decreasing rearrangement $(b_1, b_2, \dots, b_n)$ satisfies $b_i \leq i$. We denote by $\mathcal{P}_n$ the set of all parking functions of length n.*

Below we give an alternative definition of a parking function (which will explain the name "parking function"). The main strength of the definition which is about to follow is that it allows for an easy proof of one of our first results. Consider the following scenario. Suppose that n cars labelled $1, 2, \dots, n$ are trying to park in n parking spots, also labelled $1, 2, \dots, n$. Further suppose that each car has a preferred parking spot i.e. car 1 prefers to park in spot a_1, car 2 in spot a_2 and so on. Call the sequence $(a_1, a_2, \dots, a_n)$ the *preference sequence*. The cars will attempt to park in the following manner. Car 1 will park in its preferred spot. Now suppose that cars 1 to $i-1$ have parked. Car i will try to park by driving to its preferred spot and if it is unoccupied, it will park in its preferred spot. If it is occupied then it will drive to the next (in numerical order) unoccupied spot. If no such spot exists, car i cannot park. We now give the second definition of a parking function. See Figure 2.1.

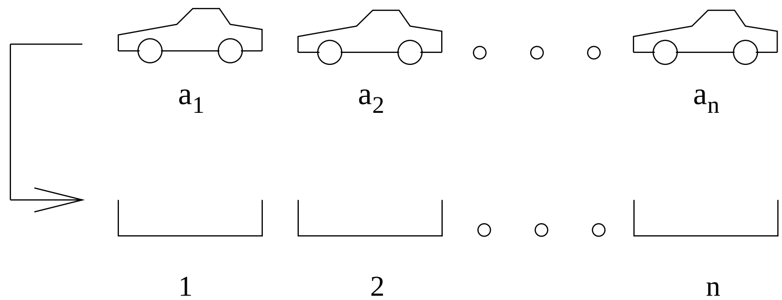

Figure 2.1: The parking scenario.

Definition 2.5 *Given the above scenario, a parking function is a preference sequence which allows all n cars to park.*

Before we proceed, let us see why the above two definitions are equivalent. To see that the first definition is a necessary condition for the second, let us assume a set of n cars has the preference sequence $(a_1, a_2, \ldots, a_n)$ that does not satisfy the first definition, i.e. if $(b_1, b_2, \ldots, b_n)$ is the non-decreasing rearrangement of $(a_1, a_2, \ldots, a_n)$ then for some k, $b_k > k$. Then the $n-k$ cars corresponding to $b_{k+1}, \ldots, b_n$ will try to park in the fewer than $n - b_k$ parking spots and since $n - b_k \leq n - k$ we see not all cars can park.

The sufficiency of the first definition for the second is similar. Suppose that the preference sequence $(a_1, a_2, \ldots, a_n)$ does not allow all cars to park. Assume that after all cars have attempted to park (the ones that can't park just leave the lot) the empty spot with the largest label is i. From this we can deduce that at least $n-i+1$ cars had preferred spots $i+1$ to n (the $n-i$ cars that are parked from spots $i+1$

to n and at least one car that left the lot must have preferred a spot greater than i for otherwise one of them would have parked in spot i). In the non-decreasing rearrangement $(b_1, b_2, \ldots, b_n)$ of $(a_1, a_2, \ldots, a_n)$, we see that $b_i, b_{i+1}, \ldots, b_n$ are all greater than or equal to $i+1$. In particular, $b_i \geq i+1 > i$.

2.3 Some Basic Results

In this section we prove a few basic results about parking functions which will motivate much of what is to come. The first result pertains to the decomposition of parking functions into smaller parking functions. We will later see that this will be a very powerful tool in proving some facts about parking functions. The second result concerns primitive parking functions (which will be defined below). In Section 2.3.1, we will count the number of parking functions.

2.3.1 Decomposition of Parking Functions

The following two propositions gives us a way to decompose parking functions and are due to Beissinger and Peled [2, Decomposition Lemma]. First we need some notation. If $(a_1, a_2, \ldots, a_n)$ is a parking function, let a_n^* be the largest integer such that $(a_1, a_2, \ldots, a_n^*)$ is still a parking function, i.e. $(a_1, a_2, \ldots, a_n^*)$ is a parking function but $(a_1, a_2, \ldots, a_n^* + 1)$ is not. Call $(a_1, a_2, \ldots, a_n^*)$ the *reduced form* of $(a_1, a_2, \ldots, a_n)$

Proposition 2.6 *Suppose that $(a_1, a_2, \ldots, a_n)$ is a parking function of length n with reduced form $(a_1, a_2, \ldots, a_n^*)$. Let $(b_1, b_2, \ldots, b_n)$ be the non-decreasing rear-*

rangement of $(a_1, a_2, \ldots, a_n^*)$. *Then, there is a unique* l *satisfying* $b_l = a_n^*$, *namely* $l = a_n^*$.

Proof. Let $l = a_n^*$. Clearly, for some i, $b_i = a_n^*$. From the definition of a parking function, we must have $i \geq l$. If $i > l$ then we can clearly increase a_n^* to at least i and still have a parking function, a contradiction. Thus, we see the only choice for i is l. $\square$

Proposition 2.7 *Let* $(a_1, a_2, \ldots, a_n)$ *be a parking function and* $(a_1, a_2, \ldots, a_n^*)$ *its reduced form with* $l = a_n^*$. *Define* $A_1 = \{i \mid a_i < a_n^*\}$ *and* $A_2 = \{i \mid a_i > a_n^*\}$. *Then both* $(a_i)_{i \in A_1}$ *and* $(a_i - l)_{i \in A_2}$ *are parking functions of length* $l - 1$ *and* $n - l$, *respectively.*

Proof. It is immediate from Definition 2.4 that $(a_i)_{i \in A_1}$ is a parking function. Using the notation from Proposition 2.6, the terms $(a_i)_{i \in A_2}$ correspond to $b_{l+1}, b_{l+2}, \ldots, b_n$ in the non-decreasing rearrangement of $(a_1, a_2, \ldots, a_n)$. Hence, $b_{l+1} \leq l + 1, b_{l+2} \leq l + 2$ and so on, implying that $(b_{l+1} - l) \leq 1, (b_{l+2} - l) \leq 2$ and so on. Hence, $(a_i - l)_{i \in A_2}$ is a parking function. $\square$

Example 2.8 For the parking function $(7, 8, 5, 2, 3, 3, 1, 2)$, we see that the largest number a_n^* such that $(7, 8, 5, 2, 3, 3, 1, a_n^*)$ is still a parking function is $a_n^* = 6$. Indeed, in the non-decreasing rearrangement of $(7, 8, 5, 2, 3, 3, 1, 6)$, which is $(1, 2, 3, 3, 5, 6, 7, 8)$, there is only one l such that $b_l = 6$ and that is $l = 6$. $\square$

2.3.2 Primitive Parking Functions

A *primitive parking function* of length n is a non-decreasing sequence of length n that is a parking function. Our next result concerns the number of primitive parking functions of length n.

Proposition 2.9 *The number of primitive parking functions of length n is the Catalan number $\frac{1}{n+1}\binom{2n}{n}$.*

Proof. It follows from Proposition 2.7 that every primitive parking function decomposes into two primitive parking functions of length $l-1$ and $n-l$ where l is also given in Proposition 2.7. Conversely, given two primitive parking functions of length $l-1$ and $n-l$ we can make a primitive parking function of length n in the obvious manner; if $a_1, a_2, \ldots, a_{l-1}$ and $b_1, b_2, \ldots, b_{n-l}$ are two primitive parking functions then $(a_1, \ldots, a_{l-1}, l, b_1+l, b_2+l, \ldots, b_{n-l}+l)$ is a primitive parking function of length n. Hence, if $f(n)$ is the number of primitive parking functions of length n, then

$$f(n) = \sum_{i=0}^{n-1} f(i)f(n-i-1)$$

Here, we note that $f(0) = f(1) = 1$. If we set $F(x) = \sum_{n=0}^{\infty} f(n)x^n$ then

$$\begin{aligned} F(x) &= \sum_{n=0}^{\infty} f(n+1)x^{n+1} + 1 \\ &= xF(x)^2 + 1 \end{aligned} \tag{2.2}$$

The last line above implies that

$$F(x) = \frac{1 - \sqrt{1 - 4x}}{2x}$$

a function which has a Taylor series

$$F(x) = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{2n}{n} x^n \tag{2.3}$$

Alternatively, we can apply the Lagrange Inversion Formula (see Goulden and Jackson [7, Sec. 1.2] Stanley [24, Sec. 5.4]) to (2.2) to obtain (2.3). This completes the proof. □

2.3.3 The Total Number of Parking Functions

We now use Definition 2.4 to prove the next result, which can be found in Foata and Riordan [5] and is due to H. Pollak.

Proposition 2.10 *The number of parking functions of length n is $(n+1)^{n-1}$.*

Proof. Suppose that we have n cars and they are going to try and park in n spots. We will modify our parking scenario slightly by adjoining an $(n+1)^{\text{st}}$ parking space to the lot and making a circular lot in such a manner that one would move in the clockwise direction to get from $n+1$ to 1. We allow the parking spot $n+1$ as a preferred spot. Let us consider preference sequences $(a_1, a_2, \ldots, a_n)$ with the property that $1 \leq a_i \leq n+1$. The cars park in the same manner that they would in the linear lot except that if a car tries to park in its preferred spot and it is occupied then the car will drive in the clockwise direction and find the next unoccupied spot.

It is clear in this scenario that all the cars can park, leaving one spot empty. It is also clear that the empty spot is the spot labelled $n+1$ if and only if the preference sequence is a parking function of length n. Further, it is clear that if the sequence $(a_1, a_2, \ldots, a_n)$ leaves spot k empty then the sequence $(a_1+i, a_2+i, \ldots, a_n+i)$ leaves spot $k+i \pmod{n+1}$ empty. The *orbit* of the sequence $(a_1, a_2, \ldots, a_n)$ is the set of sequences $\{(a_1, a_2, \ldots, a_n), (a_1+1, a_2+1, \ldots, a_n+1), \ldots, (a_1+n, a_2+n, \ldots, a_n+n)\}$. It follows from the above that given a preference sequence, where each term in the sequence is between 1 and $n+1$, then the orbit of the sequence contains precisely one parking function, the sequence that leaves spot $n+1$ empty. An easy observation is that given two preference sequences the orbits of the two sequences either coincide or are disjoint. Hence, decompose the set of all parking functions into classes of size $n+1$ with each class containing precisely one parking function. Since the number of preference sequences is $(n+1)^n$ the number of parking functions is $(n+1)^{n-1}$.□

Notes and References

§2.1 In this thesis, it will be assumed that the reader is comfortable with the objects discussed in this section, namely group representations, symmetric functions and partially ordered sets. It is also assumed that the reader is familiar with generating functions and permutation groups. The two books Ledermann [11] and Sagan [15] are similar and give a very thorough introduction to the representation of groups. Concerning symmetric function, the landmark book Macdonald [12] gives a full account of symmetric functions. Another book on the subject of symmetric functions is Stanley [24]. An advantage of [24] is that it also gives a great treatment of generating functions. Two other great sources on generating functions are Goulden and Jackson [7] and Wilf [26]. The book [7] gives a very complete account whereas [26] gives a simpler, briefer account of the material and is, currently, available free of charge from the author's website. A basic treatment of partially ordered sets is given in Stanley [20]. For those needing reference on the symmetric group, the above books Sagan [15], Lederman [11] and Macdonald [12] all deal with them since it is impossible to attack the material in those books without the symmetric group. The book Dixon and Mortimer [3] is fully devoted to the symmetric group.

§2.3 The decomposition of parking functions (Proposition 2.6 and 2.7) can be found in Beissinger and Peled [2, Decomposition Lemma]. There, the authors give a result concerning parking functions (in the paper the authors, in fact, discuss "major sequences" which are sequences $(a_1, a_2, \dots, a_n)$ such that $(n-a_1, n-a_2, \dots, n-a_n)$ are parking functions) and external activity of trees. Proposition 2.10 is proven in many different papers and in just as many different ways. It is proved in Konheim and Weiss [9], the paper that parking functions originally appeared in, using recursion. The proof given here is due to H. Pollak and is given in Foata and Riordan [5] (a paper of which Pollak is not the author). The proof can also be found in Stanley [24, Ex. 5.49].

Chapter 3

Noncrossing Partitions

Noncrossing partitions are objects that are not specific to combinatorics; they are found in many other areas of mathematics. In this chapter we explore the relationship between noncrossing partitions and parking functions. We will do this with techniques ranging from simple to more sophisticated.

3.1 Definitions and Elementary Results

We begin with the definition of a noncrossing partition. We use the notation $[n] = \{1, 2, \ldots, n\}$.

Definition 3.1 *A noncrossing partition of the set $[n]$ is a set partition of $[n]$ such that if $a < b < c < d$ and B and B' are blocks of our partition then $a, c \in B$ and $b, d \in B'$ imply that $B = B'$. We denote by NC_n the set of all noncrossing partitions of $[n]$.*

A nice way to graphically visualize a noncrossing partition of $[n]$ is by drawing $[n]$ on a circle and then for each block, draw the convex hull of the points in that block.

Example 3.2 For the partition of $[12]$ into the blocks $\{\{1, 6, 7, 10, 11\}, \{2, 4, 5\}, \{3\}, \{8, 9\}, \{12\}\}$ the graphical representation is in Figure 3.1. $\square$

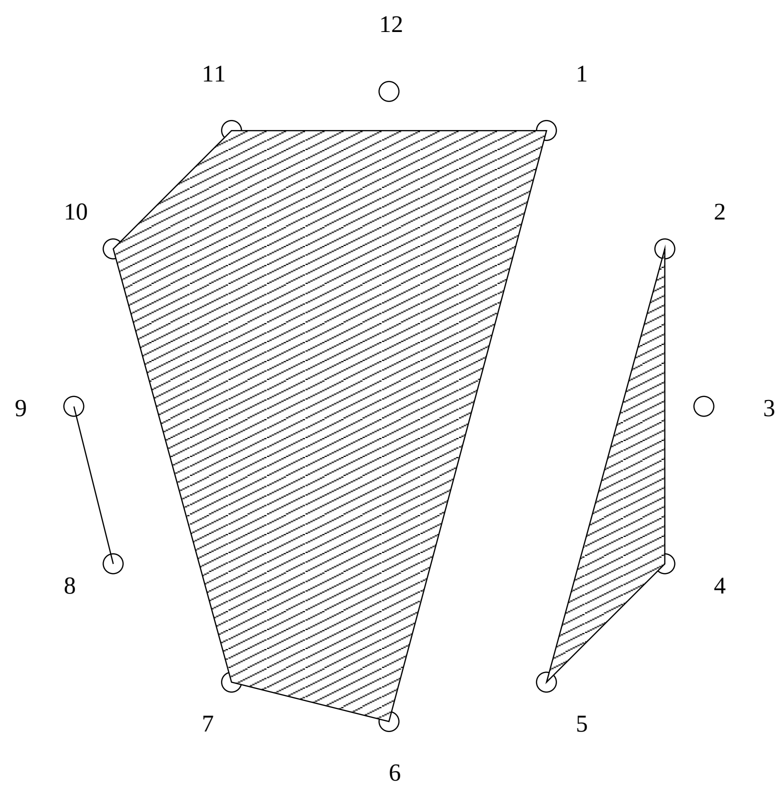

Figure 3.1: A graphical look at a noncrossing partition.

One of the properties that we will be looking at later is that the set of noncrossing partitions of length n, which we will denote by NC_n, is a poset where the ordering is given by *refinement*; that is, $\pi \leq \sigma$ if every block of π is contained in a block of σ. Notice that NC_n has both a $\hat{0}$ (which is $\{\{1\}, \{2\}, \ldots, \{n\}\}$) and a $\hat{1}$ (which is $\{1, 2, \ldots, n\}$). It is clear that every maximal chain has length n and, thus, the poset NC_n is graded and the rank function ρ is given by $\rho(\{B_1, B_2, \ldots, B_k\}) = (|B_1| - 1) + (|B_2| - 1) + \cdots + (|B_k| - 1)$. Later, we will further discuss the poset properties of noncrossing partitions (see Section 2.1.3 for poset terminology).

Our first result concerning noncrossing partitions will be the number of noncrossing partitions of length n. We will show that the number of noncrossing partitions of length n is the same as the number of primitive parking functions of length n. We will show this by demonstrating that noncrossing partitions satisfy the same recursion as primitive parking functions (and from the obvious fact that the number of noncrossing partitions of length 1 is one). To make the recursions work, we say that the number of noncrossing partitions of length 0 is also one. This, in principle, provides us with a (recursive) bijection between primitive parking functions and noncrossing partitions.

Proposition 3.3 *The number of noncrossing partitions of length n is the Catalan number $\frac{1}{n+1}\binom{2n}{n}$.*

Proof. Let c_n be the number of noncrossing partitions of length n and consider an arbitrary noncrossing partition of length n. Let k be the largest number in the block containing 1. Removing k from this block and considering all the integers less than k, we see that this can be any noncrossing partition of length $k-1$. Further, all the integers greater than k form an arbitrary noncrossing partition of length $n-k$. Thus, we see that the number of noncrossing partitions of length n and value k, where k has the property given previously, is $c_{k-1}c_{n-k}$. Summing over all possible k we get

$$c_{n+1} = \sum_{k=1}^{n} c_{k-1}c_{n-k}$$

or

$$c_{n+1} = \sum_{k=0}^{n-1} c_k c_{n-k-1}$$

Thus, noncrossing partitions satisfy the same recursion (given in Proposition 2.9) as primitive parking functions. As discussed earlier, it is clear that the number of noncrossing partitions of $[n]$ satisfies the same initial conditions that the number of primitive parking functions does. Hence, the number of noncrossing partitions of $[n]$ is the Catalan number, as claimed. $\square$

Our next result can be considered a refinement of the above result in that it shows that primitive parking functions of a certain type are in $1-\mathrm{to}-1$ correspondence with noncrossing partitions of a certain type. Of course, we must first define what we mean by a "certain type".

To each parking function $p \in \mathcal{P}_n$, we assign a *frequency sequence*, $f_1, f_2, \ldots, f_n$, where f_i is the number of i's in p. Notice that $\sum_{i=1}^{n} f_i = n$ and, therefore, we can consider the sequence $f_1, f_2, \ldots, f_n$ as a partition of n (after a suitable reordering of the sequence). We call a parking function a *parking function of type* λ if the shape of its frequency sequence is λ (see Section 2.1.1 for definition of "shape"). Similarly, we assign a frequency sequence $f_1, f_2, \ldots, f_n$ to a noncrossing partition $B = \{B_1, B_2, \ldots, B_k\}$ where $f_i = |B_i|$. We call a noncrossing partition a *noncross-*

ing partition of type λ if the sequence $f_1, f_2, \ldots, f_n$ is a partition of n. We denote the set of primitive parking functions of length n and type λ by Pr_n^λ and, similarly, the set of noncrossing partitions of $[n]$ and type λ by NC_n^λ. We will use the same notation for Pr_n^λ as we use for partitions i.e. the primitive parking function $i_1^{j_1} i_2^{j_2} \ldots i_k^{j_k}$ contains j_k occurrences of i_k.

Proposition 3.4 $|\mathrm{NC}_n^\lambda| = |\mathrm{Pr}_n^\lambda|$

Proof. Suppose that $B = \{B_1, B_2, \ldots, B_k\}$ is a noncrossing partition in NC_n^λ. Let $I = (i_1, i_2, \ldots, i_k)$ where i_j is the smallest number in B_j. We assume the blocks $B_1, B_2, \ldots, B_k$ are ordered so that $i_1 < i_2 < \cdots < i_k$. Define $\psi_n : \mathrm{NC}_n^\lambda \longrightarrow \mathrm{Pr}_n^\lambda$ by

$$\psi_n(B) = i_1^{|B_1|} i_2^{|B_2|} \cdots i_k^{|B_k|}$$

(we note that ψ_n is not a function of the partition λ). We will now show that ψ_n is a bijection.

First of all, it is not clear that ψ_n actually does what it is supposed to, i.e. it is not clear that it maps elements of NC_n^λ to Pr_n^λ. However, if we look at ψ_n a slightly different way it does become clear. Consider for each noncrossing partition B in NC_n^λ the function $\theta_B : [n] \longrightarrow [n]$ defined by

$$\theta_B(i) = \text{the smallest element of the block containing } i \text{ in } B$$

Clearly, $(\theta_B(1), \theta_B(2), \ldots, \theta_B(n))$ is just $\psi_n(B)$ with the elements of $\psi_n(B)$ permuted and $\theta_B(i) \leq i$, implying that $(\theta_B(1), \theta_B(2), \ldots, \theta_B(n))$ (and, hence, $\psi_n(B)$) is a parking function.

We now want to show that ψ_n is a bijection. We do this by defining ψ_n's inverse. To this end, we will define a function $\phi_n : \mathrm{Pr}_n^\lambda \longrightarrow \mathrm{NC}_n^\lambda$ by the following rule. We define ϕ_n recursively and set $\phi_1(1) = \{\{1\}\}$. Now suppose that $p \in \mathrm{Pr}_n^\lambda$ and that $p = i_1^{j_1} i_2^{j_2} \dots i_k^{j_k}$. We decompose p into two objects, one object being the primitive parking function $\bar{p} = i_1^{j_1} i_2^{j_2} \dots i_{k-1}^{j_{k-1}}$ and second object the ordered pair $(i_k,\ j_k)$. Recursively, we know that $\phi_{n-j_k}(\bar{p}) = \{A_1, A_2, \dots, A_k\}$ where $\{A_1, A_2, \dots, A_k\}$ is a partition of $[n - j_k]$. Define

$$\phi_n(p) = \{\bar{A}_1, \bar{A}_2, \dots, \bar{A}_{k+1}\}$$

where

$$\bar{A}_l = \{\ x \mid x \in A_l \text{ and } x < i_k\} \cup \{\ x + j \mid x \in A_l \text{ and } x \geq i_k\}$$

for $1 \leq l \leq k$ and

$$\bar{A}_{k+1} = \{i_k, i_k + 1, \dots, i_k + j_k - 1\}$$

It is clear that ϕ_n does its job, i.e., it maps primitive parking functions of length n of type λ to noncrossing partitions of length n and type λ. Further, it is clear that $\psi_n \circ \phi_n(p) = p$ for any parking function of length n. Thus, we see that ψ_n is a bijection, proving the claim. $\square$

3.2 The Noncrossing Partition Symmetric Function and the Parking Function Symmetric Function

3.2.1 Definitions and Useful Concepts

Consider the set of parking functions, $\mathcal{P}_n$, of length n. The symmetric group, $\mathfrak{S}_n$, acts on $\mathcal{P}_n$ (as a group action) in an obvious way; by permuting the coordinates of a parking function.

Example 3.5 $p = (1,3,5,2,1,5,3,2) \in \mathcal{P}_8$ and $\sigma = (132)(45) \in \mathfrak{S}_8$ we have $\sigma(p) = (3,5,1,1,2,5,3,2)$. □

We can consider the *permutation representation* given in Example 2.2 associated with the above action of $\mathfrak{S}_n$ on $\mathcal{P}_n$. That is, we consider the vector space V of all complex linear combinations of elements in $\mathcal{P}_n$, $\mathbb{C}[\mathcal{P}_n]$, and our representation $X : \mathfrak{S}_n \longrightarrow GL(V)$ will be the matrix associated with the following linear transformation: for $v = a_1v_1 + a_2v_2, \ldots, a_nv_n$ (where $v_1, v_2, \ldots, v_n$ are the n parking functions)

$$X(g)(v) = a_1 g.v_1 + a_2 g.v_2, \ldots, a_n g.v_n$$

where $g.v$ represents the above action.

Example 3.6 We can explicitly compute the case $n = 2$. In this case, the 3 parking functions (and, hence the dimension of the representation is 3) are 11, 12 and 21. The above form an ordered basis for the vector space $V = \mathbb{C}[\mathcal{P}_2]$. The representation $X : \mathfrak{S}_2 \longrightarrow GL(V)$ is given as follows. The 2 elements of the group $\mathfrak{S}_2$ are e (the identity) and $(1\,2)$. The identity element e fixes all of the basis vectors and $(1\,2).11 = 11$, $(1\,2).12 = 21$ and $(1\,2).21 = 12$. Hence, the permutation representation is given by

$$X(e) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$X((1\,2)) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

□

The *parking function symmetric function on* n, denoted as PF_n, is

$$\mathrm{PF}_n = \sum_{\mu \vdash n} z_\mu^{-1} \chi_\mu p_\mu$$

where for $\mu = (1^{m_1(\mu)}, 2^{m_2(\mu)}, \ldots, n^{m_n(\mu)})$

$$z_\mu = 1^{m_1(\mu)} m_1(\mu)!\, 2^{m_2(\mu)} m_2(\mu)! \ldots n^{m_n(\mu)} m_n(\mu)!$$

and χ_μ is the value of the character of the permutation representation above on the conjugacy class μ. This mapping, which maps characters to symmetric functions, is not particular to the above representation, but can be used on any representation. It is known as the *Frobenius characteristic* and is often (see [12, Sec. 1.7]) denoted by ch; that is, for any character χ,

$$\mathrm{ch}(\chi) = \sum_{\mu \vdash n} z_\mu^{-1} \chi_\mu p_\mu \tag{3.1}$$

Our aim is to show a connection between PF_n and F_{NC_n} (where F_p is defined in Section 2.1.3). We do this now.

3.2.2 The Parking Function Symmetric Function, PF_n

In order to learn more about PF_n we are going to learn something about the character of the above permutation representation X. To do this, we look for submodules of our representation that are easy to work with, as the characters of the submodules will sum up to the character of the entire representation (see Ledermann [11, Sec. 1.4] Sagan [15, Sec. 1.4]). Notice that the orbit of a parking function under the above group action will just consist of all possible permutations of the primitive parking function in that orbit. Thus, the representation above restricted to an orbit (under the group action) of a parking function will form a submodule. It is clear that the representation X restricted to the submodules associated with the orbits of two parking functions are equivalent representations if and only if the parking functions are of the same type. Thus, we denote the submodule corresponding to orbits of parking functions of type λ by X^λ and their associated character by χ^λ (we note that our use of χ^λ deviates from the notation in Macdonald [12]. In that book, χ^λ refers to the *irreducible* character of the symmetric group indexed by the partition λ of n.). It follows that the multiplicity of a particular submodule is the number of primitive parking functions that have type associated to that representation. The following two lemmas will be very important for computing PF_n. We refer the reader to Section 2.1.1 for the definition of h_λ, the complete symmetric function, for we will be using in the next lemma.

Lemma 3.7 (Key Lemma) *The contribution of the orbit of a parking function of type λ to PF_n is h_λ, the complete symmetric function. Phrased differently, in PF_n the contribution of all the terms that contain some fixed character χ^λ is h_λ (not counting the multiplicity of that character).*

Lemma 3.8 *The submodule associated with a parking function of type λ has multiplicity*

$$\frac{1}{n+1}\binom{n+1}{m_0(\lambda), m_1(\lambda), \ldots, m_n(\lambda)} \tag{3.2}$$

where $\binom{n+1}{m_0(\lambda), m_1(\lambda), \ldots, m_n(\lambda)}$ is the multinomial coefficient and $m_0(\lambda) = n + 1 - \sum_{i=1}^n m_i(\lambda)$. Equivalently, the number of primitive parking functions of type λ is the above number.

Notice Proposition 3.4 and Lemma 3.8 imply that the number of noncrossing partitions of type λ is given by (3.2)

Before we prove these two lemmas, we first see how they are useful in computing PF_n. The lemmas state that the contribution of a submodule corresponding to the orbit of a parking function of type λ to PF_n is h_λ. Further, the multiplicity of the submodule in the representation is

$$\frac{1}{n+1}\binom{n+1}{m_0(\lambda), m_1(\lambda), \ldots, m_n(\lambda)}$$

where $m_0(\lambda) = n+1-\sum_{i=1}^{n} m_i(\lambda)$. Let χ be the character of the permutation representation above and suppose

$$\chi = \sum_{\gamma \vdash n} c_\gamma \chi^\gamma$$

is the decomposition of χ into the above submodules indexed by partitions γ of n (which correspond to submodules associated with parking functions of type γ) and with c_γ as the multiplicity of χ^γ. Then, we have

$$\begin{aligned}
\mathrm{PF}_n &= \sum_{\mu \vdash n} z_\mu^{-1} \chi_\mu p_\mu \\
&= \sum_{\mu \vdash n} z_\mu^{-1} (\sum_{\gamma \vdash n} c_\gamma \chi^\gamma)_\mu p_\mu \\
&= \sum_{\gamma \vdash n} c_\gamma \sum_{\mu \vdash n} z_\mu^{-1} \chi_\mu^\gamma p_\mu \\
&= \sum_{\gamma \vdash n} c_\gamma \sum_{\mu \vdash n} \mathrm{ch}(\chi^\gamma) && (3.3) \\
&= \sum_{\gamma \vdash n} \frac{1}{n+1}\binom{n+1}{m_0(\gamma), m_1(\gamma), \ldots, m_n(\gamma)} h_\gamma && (3.4)
\end{aligned}$$

where $\mathrm{ch}(\chi^\gamma)$ is defined in (3.1), (3.3) follows from Lemma 3.7 and (3.4) follows from Lemma 3.8. Finally, a simple computation will reveal that (3.3) simplifies to

$$[t^n]\frac{1}{n+1}H(t)^{n+1}$$

where

$$H(t) = h_0 + h_1 t + h_2 t^2 + \dots \tag{3.5}$$

We state this a proposition for easy reference later. It appears in Stanley [23, (1)].

Proposition 3.9

$$\mathrm{PF}_n = [t^n]\frac{1}{n+1}H(t)^{n+1}$$

It is now time to prove the above two lemmas.

Proof of Lemma 3.7. Suppose that χ^λ is the character of a submodule corresponding to a primitive parking function p of type λ and let Ω be the orbit of p. Our goal is to show that

$$\sum_{\mu \vdash n} z_\mu^{-1} \chi_\mu^\lambda p_\mu = h_\lambda \tag{3.6}$$

It is more instructive to begin by working backwards i.e. show that

$$h_\lambda = \sum_{\mu \vdash n} z_\mu^{-1} \chi_\mu^\lambda p_\mu \tag{3.7}$$

(it turns out doing this motivates our steps more). It is well known that

$$h_k = \sum_{\mu \vdash k} z_\mu^{-1} p_\mu$$

(see Macdonald [12, (2.14′)]) for any k and, hence,

$$h_\lambda = \left(\sum_{\mu \vdash \lambda_1} z_\mu^{-1} p_\mu\right)\left(\sum_{\mu \vdash \lambda_2} z_\mu^{-1} p_\mu\right)\cdots\left(\sum_{\mu \vdash \lambda_n} z_\mu^{-1} p_\mu\right)$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$. The right hand side of the above is, by definition

$$\mathrm{ch}(1_{\mathfrak{S}_{\lambda_1}}) \cdot \mathrm{ch}(1_{\mathfrak{S}_{\lambda_2}}) \cdots \mathrm{ch}(1_{\mathfrak{S}_{\lambda_n}})$$

where $1_{\mathfrak{S}_m}$ is the trivial character on the group $\mathfrak{S}_m$ and this is

$$\mathrm{ch}(\mathrm{ind}_{\mathfrak{S}_{\lambda_1} \times \mathfrak{S}_{\lambda_2} \times \ldots \mathfrak{S}_{\lambda_n}}^{\mathfrak{S}_{\lambda_1+\lambda_2+\cdots+\lambda_n}} 1_{\mathfrak{S}_{\lambda_1}} \times 1_{\mathfrak{S}_{\lambda_2}} \times \ldots \times 1_{\mathfrak{S}_{\lambda_n}}) \tag{3.8}$$

(see Macdonald [12, (7.3)]) Let us see what (3.8) really means. Let us begin with the character $(1_{\mathfrak{S}_{\lambda_1}} \times 1_{\mathfrak{S}_{\lambda_2}} \times \ldots \times 1_{\mathfrak{S}_{\lambda_n}})$. Given the way that we embed $\mathfrak{S}_{\lambda_1} \times \mathfrak{S}_{\lambda_2} \times \ldots \times \mathfrak{S}_{\lambda_n}$ into $\mathfrak{S}_{\lambda_1+\lambda_2+\ldots+\lambda_n}$ we see that $(1_{\mathfrak{S}_{\lambda_1}} \times 1_{\mathfrak{S}_{\lambda_2}} \times \ldots \times 1_{\mathfrak{S}_{\lambda_n}})$ is simply $1_{\mathfrak{S}_\lambda}$, where $\mathfrak{S}_\lambda$ is the *Young Subgroup* (see Section 2.1.2 for the definition). Hence, we want to compute

$$\mathrm{ch}(\mathrm{ind}_{\mathfrak{S}_{\lambda_1}\times\mathfrak{S}_{\lambda_2}\times\ldots\mathfrak{S}_{\lambda_n}}^{\mathfrak{S}_{\lambda_1+\lambda_2+\cdots+\lambda_n}} 1_{\mathfrak{S}_\lambda}) \tag{3.9}$$

Let $[1]_{\mathfrak{S}_\lambda}$ be the representation associated with the above trivial character $1_{\mathfrak{S}_\lambda}$. Let $t_1, t_2, \ldots, t_m$ be a *transversal* (see Section 2.1.2 for definition) for the subgroup $\mathfrak{S}_\lambda$. From the definition of an *induced representation* (see Section 2.1.2 for definition), we see that (3.9) is the character for the representation, $\bar{X}$, given by

$$\bar{X} = \mathrm{ind}_{\mathfrak{S}_\lambda}^{\mathfrak{S}_n}[1]_{\mathfrak{S}_\lambda}(g) = \begin{bmatrix} [1]_{\mathfrak{S}_\lambda}(t_1 g t_1^{-1}) & [1]_{\mathfrak{S}_\lambda}(t_1 g t_2^{-1}) & \cdots & [1]_{\mathfrak{S}_\lambda}(t_1 g t_m^{-1}) \\ [1]_{\mathfrak{S}_\lambda}(t_2 g t_1^{-1}) & [1]_{\mathfrak{S}_\lambda}(t_2 g t_2^{-1}) & \cdots & [1]_{\mathfrak{S}_\lambda}(t_2 g t_m^{-1}) \\ \vdots & \vdots & \ddots & \vdots \\ [1]_{\mathfrak{S}_\lambda}(t_m g t_1^{-1}) & [1]_{\mathfrak{S}_\lambda}(t_m g t_2^{-1}) & \cdots & [1]_{\mathfrak{S}_\lambda}(t_m g t_m^{-1}) \end{bmatrix}$$

where in the above matrix $[1]_{\mathfrak{S}_\lambda}(g) = 0$ if $g \notin \mathfrak{S}_\lambda$. Notice that the (i, i) entry on the main diagonal of the above matrix (in fact, any entry) will be equal to 1 if $t_i g t_i^{-1} \in \mathfrak{S}_\lambda$ and 0 if it is not. Hence, the trace of the above matrix will be the number of $t_i g t_i^{-1}$ on the main diagonal that lie in the Young Subgroup $\mathfrak{S}_\lambda$. Therefore, the character of the above representation, $\bar{\chi}$, is

$$\bar{\chi}(g) = \#\{\, i \mid t_i g t_i^{-1} \in \mathfrak{S}_\lambda\} \tag{3.10}$$

Call the right hand side of (3.10), $N(g)$. The claim is that the above character is equal to the character χ^λ (and, therefore, by the uniqueness of characters (see [15, Cor. 1.9.4]), the representation $\bar{X}$ and X above are the same). By proving the above claim we will prove that (3.7) holds, completing the proof of the lemma. We prove the claim now by showing that $N(g) = \chi(g)$.

Fix a g in G and suppose that g is in the conjugacy class C. Looking at the definition of $N(g)$, it is clear that $N(g)$ is independent of the transversal that we pick. Thus, $N(g)$ can be obtained by conjugating g by *all* the elements of $\mathfrak{S}_n$ and then dividing by the number of transversals i.e.

$$N(g) = \frac{|\{\ t \mid t \in \mathfrak{S}_n \text{ and } tgt^{-1} \in \mathfrak{S}_\lambda\}|}{\mathfrak{S}_\lambda}$$

We note that every element in the conjugacy class C will appear the same number of times as a conjugate of g, that is, if a and b are both in C then the number of conjugates of g that are equal to a is the same as the number of conjugates of g that are equal to b. Therefore, conjugating g by all the $|\mathfrak{S}_n|$ elements of $\mathfrak{S}_n$, each conjugate of g will appear $|\mathfrak{S}_n|/|C|$ times. But of these, only $|\mathfrak{S}_\lambda \cap C|$ are in $\mathfrak{S}_\lambda$. Therefore, we see that

$$|\{\ t \mid t \in \mathfrak{S}_n \text{ and } tgt^{-1} \in \mathfrak{S}_\lambda\}| = |\mathfrak{S}_\lambda \cap C|\frac{|\mathfrak{S}_n|}{|C|}$$

implying that

$$N(g) = \frac{|\{\ t \mid t \in \mathfrak{S}_n \text{ and } tgt^{-1} \in \mathfrak{S}_\lambda\}|}{|\mathfrak{S}_\lambda|} = \frac{|\mathfrak{S}_\lambda \cap C|}{|\mathfrak{S}_\lambda|}\frac{|\mathfrak{S}_n|}{|C|} \tag{3.11}$$

We make the easy observation that the number of parking functions in Ω is

$$\binom{n}{m_1(\lambda), m_2(\lambda), \ldots, m_n(\lambda)} \tag{3.12}$$

and it also clear that $|\mathfrak{S}_n|/|\mathfrak{S}_\lambda|$ is equal to (3.12). Thus, we see that from (3.11) that

$$N(g) = \frac{|\mathfrak{S}_\lambda \cap C|}{|C|}|\Omega|.$$

Since $\mathfrak{S}_n$ acts *transitively* (see Section 2.1.2 for definition) on Ω we have that $N(g) = \mathrm{Fix}(g)$ (see Dixon and Mortimer [3, Ex. 1.7.6]), where $\mathrm{Fix}(g)$ is the number of elements of Ω fixed by g. From the definition of $\chi^\lambda(g)$ we see that $\chi^\lambda(g) = \mathrm{Fix}(g)$ implying that $N(g) = \chi^\lambda(g)$, completing the proof. □

Proof of Lemma 3.8. In this proof we will use the parking lot definition of a parking function, i.e. Definition 2.5. Further, we will use the circular parking lot scenario we used in the proof of Proposition 2.10. The number of ways of choosing a preference sequence whose terms come from $[n+1]$ and are of "type λ" (we will take this to be defined analogously to the definition of parking functions of type λ) is clearly

$$\binom{n+1}{m_0(\lambda), m_1(\lambda), \ldots, m_n(\lambda)}$$

However, we found in Proposition 2.10 that preference sequences of length n whose terms come from $[n+1]$ are in $(n+1) - \text{to} - 1$ correspondence with parking functions of length n. This correspondence clearly holds when we restrict the above to primitive parking functions of type λ; that is, preference sequences of length n whose terms come from $[n+1]$ and are of type λ are in $(n+1)-\text{to}-1$ correspondence with primitive parking functions of length n of type λ. Hence, the number of primitive parking functions of type λ is

$$|\mathrm{Pr}_n^{\lambda}| = \frac{1}{n+1}\binom{n+1}{m_0(\lambda), m_1(\lambda), \ldots, m_n(\lambda)}.$$

$\square$

3.2.3 Consequences of the Computation of $\mathbf{PF}_n$

The following are symmetric function expansions of PF_n, all of which can be found in Stanley [23, Prop. 2.2].

Proposition 3.10 *The following are expansions of* PF_n.

$$\mathrm{PF}_n = \sum_{\lambda\vdash n} \frac{1}{n+1}\binom{n+1}{m_0(\lambda), m_1(\lambda), \ldots, m_n(\lambda)} h_\lambda \tag{3.13}$$

$$= \sum_{\lambda\vdash n} (n+1)^{l(\lambda)-1} z_\lambda^{-1} p_\lambda \tag{3.14}$$

$$= \sum_{\lambda\vdash n} \frac{1}{n+1}\left[\prod_i \binom{n+\lambda_i}{n}\right] m_\lambda \tag{3.15}$$

$$= \sum_{\lambda\vdash n} \frac{1}{n+1} s_\lambda(1^{n+1}) s_\lambda \tag{3.16}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$. We also have

$$\sum_{n\geq 0} \mathrm{PF}_n t^{n+1} = (tE(-t))^{<-1>} \tag{3.17}$$

where $E(t) = \sum_{i\geq 0} e_i t^i$ (see Section 2.1.1 for definition of e_i) and $^{<-1>}$ denotes compositional inverse

Proof. (3.13) follows from Lemmas 3.7 and 3.8. From the definition of $H(t)$ given in (3.5) we see that

$$H(t) = \prod_{i=1}^{\infty} \frac{1}{1 - x_i t}$$

and, therefore, $H(t)$ is clearly the Cauchy product, defined as

$$\prod(\mathbf{x}, \mathbf{y}) = \prod_{i,j} \frac{1}{1 - x_i y_j} \tag{3.18}$$

with $y_1, y_2, \ldots, y_{n+1}$ equal to t and $y_i = 0$ for all $i > n + 1$. Using the two well known expansions for the Cauchy product

$$\prod(\mathbf{x}, \mathbf{y}) = \sum_{\lambda} z_\lambda^{-1} p_\lambda(\mathbf{x}) p_\lambda(\mathbf{y}) \tag{3.19}$$

and

$$\prod(\mathbf{x}, \mathbf{y}) = \sum_{\lambda} m_\lambda(\mathbf{x}) h_\lambda(\mathbf{y}) \tag{3.20}$$

(see Macdonald [12, Sec. 1.4]) we see that (3.19) implies that

$$\mathrm{PF}_n = \sum_\lambda z_\lambda^{-1} p_\lambda(1^{n+1}) p_\lambda(\mathbf{x})$$

By definition, the power sum symmetric function, p_j, is

$$p_j = x_1^j + x_2^j + \dots$$

and, hence, $p_j(1^{n+1}) = n+1$, from which (3.14) follows. (3.20) implies that

$$\mathrm{PF}_n = \sum_\lambda h_\lambda(1^{n+1})\, m_\lambda(\mathbf{x})$$

By definition,

$$h_j = \sum_{1 \le i_1 \le i_2 \le \dots \le i_j \le j} x_{i_1} x_{i_2} \dots x_{i_j}$$

and, hence, we see that $h_j(1^{n+1})$ is just the number of multisets of $[n+1]$ of size j which is well known to be the number $\binom{n+j}{n}$, from which (3.15) follows. (3.16) is a direct consequence of the well known expansion for the Cauchy identity

$$\prod_\lambda s_\lambda(\mathbf{x}) s_\lambda(\mathbf{y})$$

The final equation, (3.17), can be obtained by applying the Lagrange Inversion Formula (see Goulden and Jackson [7, Sec. 1.2] Stanley [24, Sec. 5.4]) and using the fact that

$$E(t) = \frac{1}{H(-t)}$$

(see Macdondald [12, (2.6)]). In detail,

$$\begin{aligned}
[t^{n+1}](tE(-t))^{<-1>} &= \frac{1}{n+1}[t^n]\left(\frac{1}{E(-t)}\right)^{n+1} \\
&= \frac{1}{n+1}[t^n]H(t)^{n+1} \\
&= \mathrm{PF}_n
\end{aligned}$$

□

Some things to note are interpretations for the formulas in Proposition 3.10. From the definition of PF_n and (3.14) we see for any $g \in \mathfrak{S}_n$ whose cycle type is λ that

$$\chi(g) = (n+1)^{l(\lambda)-1}$$

implying that the number of parking functions that g fixes is $(n+1)^{l(\lambda)-1}$. Further, on a thought that purely concerns symmetric functions, it is known that the Frobenius characteristic takes irreducible character χ^λ (here, we stick with the notation in Macdonald [12]. That is, χ^λ denotes the irreducible representation of the symmetric group.) indexed by partitions λ of n, to s_λ. Thus, from (3.16) we see that the multiplicity of the irreducible representation χ^λ is $s_\lambda(1^{n+1})$.

Example 3.11 We look at the simplest non-trivial example for an illustration of the above interpretation of Proposition 3.10. For $n = 3$, there are sixteen parking functions and they are

111,

112, 121, 211,

113, 131, 311,

122, 212, 221,

123, 213, 231, 321, 312, 132

and, for ease, we list them with the first parking function of each line being the primitive parking function in that orbit. We see that

$$[t^3]\frac{1}{4}H(t)^4 = h_3 + 3h_{12} + h_{111} \tag{3.21}$$

Indeed, there is one primitive parking function of type 111, three of type 12 and one of type 3. Further, from (3.14) of Proposition 3.10 we see (3.21) is also equal to

$$16z_{111}^{-1}p_{111} + 4z_{12}^{-1}p_{12} + z_3^{-1}p_3$$

The only element of $\mathfrak{S}_3$ of type 111 is the identity permutation and it, clearly, fixes all sixteen parking functions. A permutation of type 12 is $(1\,2)$ and it fixes the parking functions 111, 112, 113 and 221 for a total of 4. It is clear that any three cycle fixes only 111. $\square$

3.2.4 The Noncrossing Partition Symmetric Function

We finally make a connection between the parking function symmetric function and noncrossing partition. Much of what follows will be given without proof, however, we refer the reader to the literature containing the proofs. We apply (2.1) to the poset NC_{n+1}. The first question we wish to answer is whether or not $F_{\mathrm{NC}_{n+1}}$ is a symmetric function. According to Stanley [21, Thm. 1.4], $F_{\mathrm{NC}_{n+1}}$ is a symmetric function if NC_{n+1} is locally rank symmetric (see Section 2.1.3 for poset terminology). However, in Simion and Ullman [18, Thm. 1.1] it is shown that NC_i is self-dual and in Nica and Speicher [14, Sec. 1.3] it is shown that every interval in NC_{n+1} is a product of NC_i's. As was discussed in Section 2.1.3, this is sufficient to conclude that NC_{n+1} is locally rank symmetric. Hence, $F_{\mathrm{NC}_{n+1}}$ is a symmetric function. In Proposition 2.3, we saw that

$$F_P(x) = \sum_{\lambda \vdash n} \alpha_P(S_\lambda) m_\lambda$$

where $S_\lambda = \{\lambda_1, \lambda_1 + \lambda_2, \ldots, \lambda_1 + \lambda_2 + \cdots + \lambda_l\}$ and l is the length of λ. From the evaluation of $\alpha_{\mathrm{NC}_{n+1}}(S_\lambda)$ given in Edelman [4, Thm. 3.2] we have

$$F_{\mathrm{NC}_{n+1}} = \sum_{\lambda \vdash n} \frac{1}{n+1} \left[\prod_i \binom{n+1}{\lambda_i} \right] m_\lambda$$

The proof of the following proposition can be found in Stanley [23, Prop. 2.2].

Proposition 3.12

$$\omega \mathrm{PF}_n = \sum_{\lambda \vdash n} \frac{1}{n+1} \left[\prod_i \binom{n+1}{\lambda_i} \right] m_\lambda \tag{3.22}$$

where ω is the standard involution on the ring of symmetric functions (defined in Section 2.1.1).

Proof. We denote by ω_x the involution ω acting only on the variables $x_1, x_2, \ldots$. If we are to take the generating function

$$E(t) = \sum_{i \geq 0} e_i t^i$$

then it easily follows from the definition of e_r that

$$E(t) = \prod_i (1 + x_i t).$$

Clearly from the definition of ω and the fact that $H(t) = \prod_i \frac{1}{1-x_i t}$ we have

$$\omega H(t) = E(t) = \prod_i (1 + x_i t)$$

Also, it is clear that

$$\prod(\mathbf{x}, \mathbf{y}) = \prod_j H(y_j)$$

and, hence, we see from (3.18) that

$$\omega_x \prod(\mathbf{x}, \mathbf{y}) = \prod_j \omega_x H(y_j) = \prod_j E(y_j) = \prod_{i,j}(1 + x_i y_j). \tag{3.23}$$

Hence, we see that $\omega_x \prod(\mathbf{x}, \mathbf{y})$ is symmetric in $\mathbf{x}$ and $\mathbf{y}$. From (3.20) and the symmetry of the Cauchy product we obtain

$$\prod(\mathbf{x}, \mathbf{y}) = \sum_\lambda h_\lambda(\mathbf{x}) m_\lambda(\mathbf{y})$$

and, therefore,

$$\omega_x \prod(\mathbf{x}, \mathbf{y}) = \sum_\lambda e_\lambda(\mathbf{x}) m_\lambda(\mathbf{y})$$

Since $\omega_x \prod_{i,j}(\mathbf{x}, \mathbf{y})$ is symmetric in $\mathbf{x}$ and $\mathbf{y}$

$$\omega_x \prod_{i,j}(\mathbf{x}, \mathbf{y}) = \sum_\lambda m_\lambda(\mathbf{x}) e_\lambda(\mathbf{y})$$

Therefore, $\frac{1}{n+1}[t^n]H(t)^{n+1}$ is

$$\sum_{\lambda \vdash n} \frac{1}{n+1} e_\lambda(1^{n+1}) m_\lambda$$

The definition of e_r gives $e_r(1^{n+1}) = \binom{n+1}{r}$ and the result follows. $\square$

Corollary 3.13

$$\omega \mathrm{PF}_n = F_{\mathrm{NC}_{n+1}}(x)$$

Recall from Section 3.2.1 that a noncrossing partition of type $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ is a noncrossing partition with block sizes $\lambda_1, \lambda_2, \ldots$. In the next corollary, we use the following noncrossing partition analogue of the exponential formula due to Speicher [19, pg. 616] (see also [24, Ex. 5.35]) (the following proof is the one used in [24, Ex. 5.35]). Given a function $f : \mathbb{N} \longrightarrow R$ (where R is a commutative ring with identity) with $f(0) = 1$ and $F(t) = \sum_{n \geq 0} f(n)t^n$, define the function g with $g(0) = 1$

$$g(n) = \sum_{\{B_1, B_2, \ldots, B_n\} \in \mathrm{NC}_n} f(|B_1|)f(|B_2|) \ldots f(|B_n|)$$

Then,

$$\sum_{n \geq 0} g(n)t^{n+1} = \left(\frac{t}{F(t)}\right)^{<-1>}$$

where the $^{<-1>}$ denotes the compositional inverse. The proof of this follows from Lemma 3.8 and the fact that the number of noncrossing partitions of type λ is given by the number in Lemma 3.8. Letting s_i to be the number of blocks of size i in a noncrossing partition (where $k = \sum_i s_i$), we have

$$
\begin{aligned}
g(n) &= \sum_{\{B_1,B_2,\dots,B_n\}\in \mathrm{NC}_n} f(|B_1|)f(|B_2|)\dots f(|B_n|) \\
&= \sum_{s_1+2s_2+\dots+ns_n} \frac{n!}{(n-k+1)!s_1!s_2!\dots s_n!} f(1)^{s_1} f(2)^{s_2}\dots f(n)^{s_n} \\
&= \sum_{k\geq 1} \frac{n!}{(n-k+1)!k!} \sum_{i_1+i_2+\dots+i_k=n} \frac{f(i_1)f(i_2)\dots f(i_k)}{i_1!i_2!\dots i_k!} \\
&= \frac{1}{n+1}\sum_{k\geq 1}\binom{n+1}{k}[x^n]\left(\sum_{i\geq 1} f(i)\frac{x^i}{i!}\right)^k \\
&= [x^n]\frac{1}{n+1}(F(x)^{n+1}-1) \\
&= [x^n]\frac{1}{n+1}F(x)^{n+1}
\end{aligned}
$$

where the second line above follows from Lemma 3.8. Applying the Lagrange Inversion Formula (see Goulden and Jackson [7, Sec. 1.2] Stanley [24, Sec. 5.4]), the last line above implies

$$g(n) = [x^{n+1}]\left(\frac{x}{F(x)}\right)^{<-1>}$$

The following corollary is due to Stanley [23, Prop. 2.4].

Corollary 3.14 *Let λ be a partition of n. Then the coefficient of h_λ in PF_n is equal to the number of noncrossing partitions of type λ in NC_n.*

Note the interesting fact that Corollary 3.13 refers to NC_{n+1} and Corollary 3.14 refers to NC_n.

Proof. If we set $f(n) = h_n$, then writing $g(n) = \sum_{\lambda \vdash n} u_\lambda h_\lambda$ we clearly have that u_λ is the number of members of NC_n of type λ. But if $f(n) = h_n$ then $F(t) = H(t)$ and, hence,

$$\sum_{n \geq 0} g(n) t^{n+1} = (tE(-t))^{<-1>}$$

(using the fact that $E(-t) = 1/H(t)$). But (3.17) then implies that $g(n) = \mathrm{PF}_n$ and we have our result. □

Notice that Corollary 3.14 gives us another proof of Proposition 3.4.

Notes and References

§3.1 The material in this section is due to the author of this thesis.

§3.2 The material in this section is almost wholly due R. Stanley. The work concerning the parking function symmetric function and the noncrossing partition symmetric function can be found in Stanley [23]. In it, however, Stanley calls the proof of Proposition 3.9 "clear" and omits it entirely. The details (Lemma 3.7 and 3.8) were filled in by the author of this thesis. However, Stanley does provide full proofs of Proposition 3.10 and Corollary 3.14. The proof of Corollary 3.13 is presented essentially the way it is here.

Chapter 4

Hyperplane Arrangements

A *hyperplane arrangement* in $\mathbb{R}^n$ is a discrete set of hyperplanes in $\mathbb{R}^n$. Given a hyperplane arrangement $\mathcal{H}$ in $\mathbb{R}^n$, we can remove $\mathcal{H}$ from $\mathbb{R}^n$ and we obtain a collection of connected sets. Call these connected sets the *regions* of the hyperplane arrangement $\mathcal{H}$. The question that we will primarily be interested in is the number of regions created when we remove a hyperplane arrangement from $\mathbb{R}^n$. A generating function that we will be considering is the *distance enumerator*, $D_{\mathcal{H}}$, which we now define. For the hyperplane arrangement $\mathcal{H}$ we pick a region, R_0 and call R_0 the *base region.* For any region R define $d(R)$ to be the number of hyperplanes separating (in the topological sense) R from R_0. Finally, define

$$D_{\mathcal{H}}(q) = \sum_{R} q^{d(R)} \tag{4.1}$$

where the sum is over all regions of $\mathcal{H}$. We will primarily be interested in two arrangements, the *braid arrangement* and the *Shi arrangement*, both of which will be dealt with in the following sections.

4.1 The Braid Arrangement

The *braid arrangement* in $\mathbb{R}^n$ is the set of hyperplanes $x_i - x_j = 0$ for all $1 \leq i < j \leq n$. We denote the braid arrangement by $\mathcal{B}_n$.

Example 4.1 In Figure 4.1 we have the braid arrangement for $n = 3$. Of course, the intersection of all three planes is the line $x_1 = x_2 = x_3$. $\square$

Notice that because our hyperplanes are all of the form $x_i - x_j = 0$, $1 \leq i < j \leq n$, the regions of the braid arrangement do not contain any points $(a_1, a_2, \ldots, a_n)$ such that $a_i = a_j$ for some i and j. Hence, for every $(a_1, a_2, \ldots, a_n)$ in $\mathbb{R}^n$ there exists a unique $\omega \in \mathfrak{S}$ such that $a_{\omega(1)} > a_{\omega(2)} > \ldots > a_{\omega(n)}$. The claim is that every region can be labelled with a unique permutation ω determined as above. To see this, first notice that any other point $(b_1, b_2, \ldots, b_n)$ in the same region as $(a_1, a_2, \ldots, a_n)$ also satisfies $b_{\omega(1)} > b_{\omega(2)} > \ldots > b_{\omega(n)}$. This is clear because if for some i and j, we have $a_{\omega(i)} > a_{\omega(j)}$ and $b_{\omega(j)} > b_{\omega(i)}$ then $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ lie on opposite sides of the hyperplane $x_{\omega(i)} - x_{\omega(j)} = 0$ and, hence, cannot be in the same region. Furthermore, for any two permutations ω and ω' the two regions labelled by ω and ω' must be different. This is because of the fact that there must exist an i and j such that $\omega(i) > \omega(j)$ but $\omega'(i) < \omega'(j)$ (to see why different permutations have this property see Proposition 4.4 below). In this case, the region labelled ω would be on the side $x_{\omega(i)} - x_{\omega(j)} > 0$ of the hyperplane $x_{\omega(i)} - x_{\omega(j)} = 0$ and the region labelled ω' would be on the side $x_{\omega(i)} - x_{\omega(j)} < 0$ of the

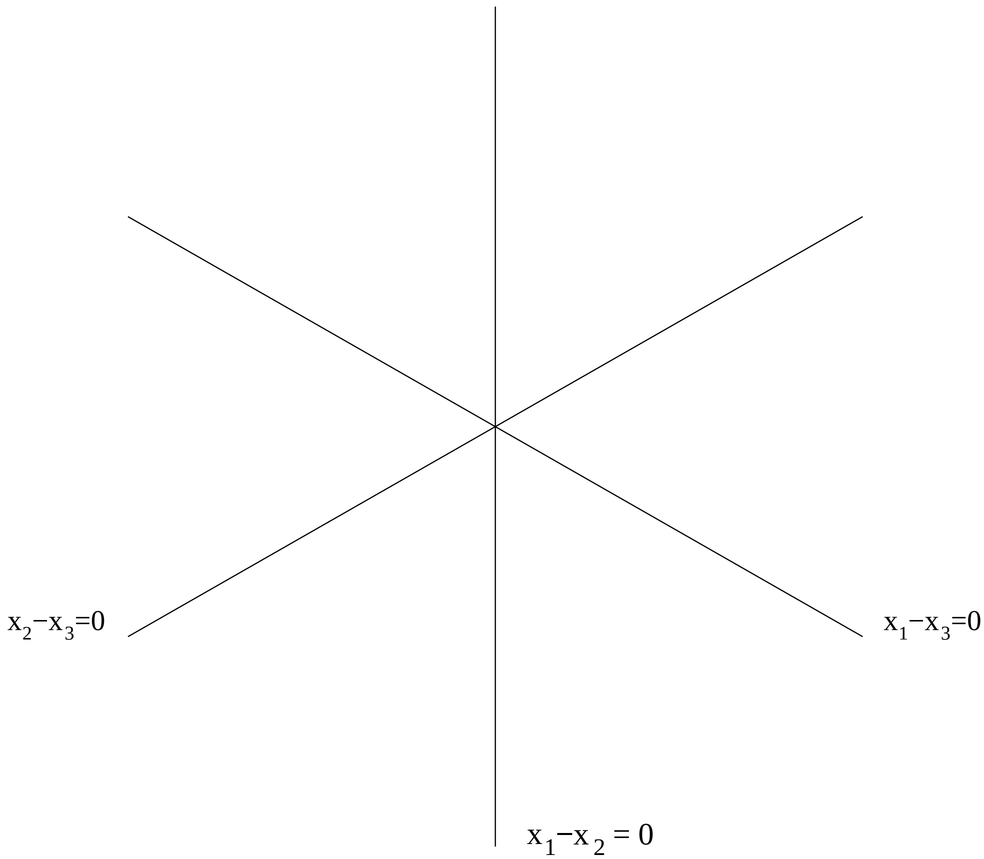

Figure 4.1: The braid arrangement for $n = 3$.

same hyperplane. Therefore, we see every region R can be uniquely labelled by the $\omega \in \mathfrak{S}_n$, where ω is the unique permutation such that every point $(a_1, a_2, \ldots, a_n)$ satisfies $a_{\omega(1)} > a_{\omega(2)} > \ldots > a_{\omega(n)}$.

Notice that the region labelled by the identity permutation contains points $(a_1, a_2, \ldots, a_n)$ such that $a_1 > a_2 > \ldots > a_n$. Let R_0 be the region $x_1 > x_2 \ldots > x_n$ labelled by the identity permutation. For any permutation ω, an *inversion* of ω is an ordered pair $(\omega(i), \omega(j))$ such that $i < j$ and $\omega(i) > \omega(j)$. From the above discussion, given any permutation ω that labels the region R, $(\omega(i), \omega(j))$ is not an inversion ω if and only if R and R_0 lie on the same side of the hyperplane $x_{\omega(i)} - x_{\omega(j)} = 0$. Hence, the number of hyperplanes that separate the region R and R_0 is the number of inversions in ω. We state this as a proposition for ease of reference later.

Proposition 4.2 *In the above labelling of $\mathcal{B}_n$ if the region R is labelled by the permutation ω then $(\omega(i), \omega(j))$ is an inversion if and only if the hyperplane $x_i - x_j = 0$ separates R from R_0.*

Proposition 4.2 can be found in Stanley [25, Intro.].

Example 4.3 The labelling discussed above for $\mathcal{B}_3$ is given in Figure 4.2. $\square$

We give a slightly different definition of the labels on the braid arrangement that will give us a particularly nice result about the generating function $D_{\mathcal{B}_n}(q)$ given in (4.1). We associate an n-tuple $\lambda(R)$ of positive integers with every region of $\mathcal{B}_n$. First, we define $\lambda(R_0) = (1, 1, \ldots, 1)$. Now suppose that $\lambda(R)$ is known and $\lambda(R')$ is not, R and R' are only separated by $x_i - x_j = 0$ and R and R_0 are on the same side of $x_i - x_j = 0$. Then, define $\lambda(R') = \lambda(R) + e_i$ where e_i is the i^{th} standard basis vector. Hence, if R is the region separated from R_0 by the set of hyperplanes S and $(a_1, a_2, \ldots, a_n)$ labels R then

$$a_i = \#\{\ i \mid x_i = x_j \in S\}$$

Notice that the same region R characterized by the set S would have been labelled, in the previous labelling, by the permutation ω that has the property that for a fixed j the number of $\omega(i)$ such that $\omega(i) > \omega(j)$ and $i < j$ is equal to $a_{\omega(j)}$ (because

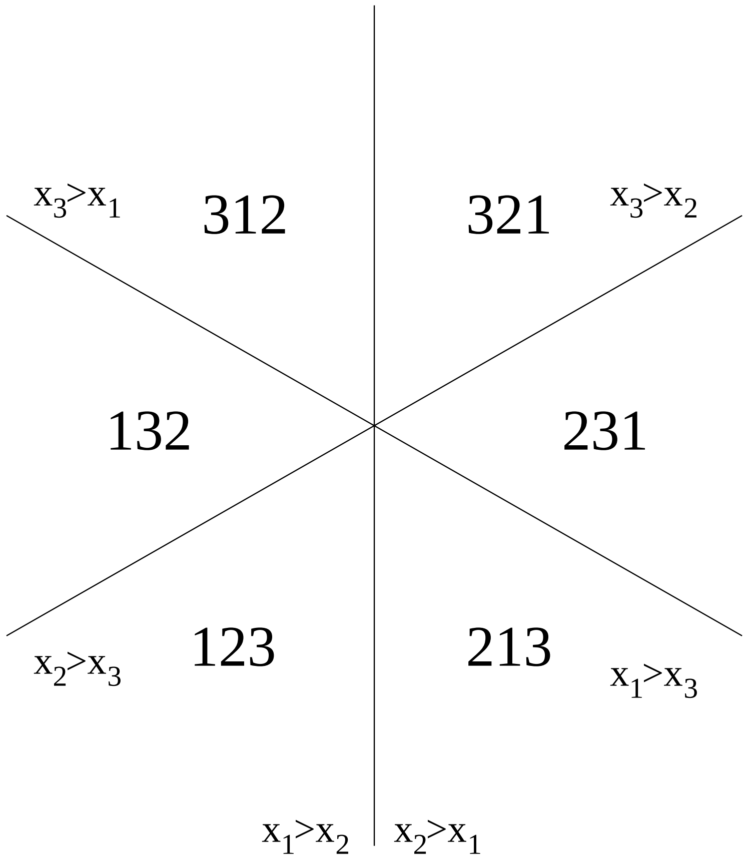

Figure 4.2: The permutation labelling of the braid arrangement.

all the hyperplanes in S of the form $x_{\omega(j)} - x_k = 0$ will form inversions with $\omega(j)$), so

$$a_{\omega(j)} = \#\{\, \omega(i) \mid \omega(i) > \omega(j) \text{ and } i < j\} \tag{4.2}$$

The sequence $(a_1, a_2, \ldots, a_n)$ in (4.2) is called the *inversion table* of ω. From an inversion table $(a_1, a_2, \ldots, a_n)$ one can construct a permutation ω (written as a word) via the following rule. Begin by starting the word as "n". Now suppose that $n, n-1, \ldots, n-i+1$ have been inserted into the word. Insert $n-i$ into the word so that there are a_{n-i} numbers to the left of it. It is immediately clear that $(a_1, a_2, \ldots, a_n)$ is the inversion table of ω. Further, it is clear that one can construct a permutation given any sequence $(a_1, a_2, \ldots, a_n)$ of non-negative integers such that $a_i \leq n-i$. What may not be immediately clear is the following proposition, which can be found in Stanley [20, Prop. 1.3.9].

Proposition 4.4 *Let $I : [0, n-1] \times [0, n-2] \times \ldots \times [0, 0] \longrightarrow \mathfrak{S}_n$ be the mapping given above. Then I is a bijection.*

Proof. To prove this we display the inverse of I. Given a permutation (written as a word) $\omega = \omega_1\omega_2 \ldots \omega_n$ we construct the inversion table as follows. a_1 is the number of entries to the left of 1 in ω. Remove 1 from the word ω. After $a_1, a_2, \ldots, a_i$ have been defined and $1, 2, \ldots, i$ removed from the word ω, set a_{i+1} to be the number of elements to the left of $i+1$ in what remains of ω. Clearly, we get an object in $[0, n-1] \times [0, n-2] \times \ldots \times [0, 0]$. Further, it is clear that the procedure outlined will be the inverse of I, completing the proof. □

Example 4.5 For $(5, 7, 3, 2, 3, 3, 0, 1, 0) \in [0, 8] \times [0, 7] \times \ldots \times [0, 0]$ the permutation ω built up via the above algorithm is

9

98

798

7986

79856

794856

7943856

79438562

794381562 □

In (4.2) $\lambda(R)$ is the inversion table of ω, where ω is the permutation labelling of R. In addition, it is clear that if $\lambda(R) = (a_1, a_2, \dots, a_n)$ then $d(R) = a_1 + a_2 + \dots + a_n$. Hence, we have the following result for the generating function $D_{\mathcal{B}_n}$ given in (4.1).

Corollary 4.6

$$D_{B_n}(q) = (1+q)(1+q+q^2)\dots(1+q+\dots+q^{n-1}), n \geq 1 \tag{4.3}$$

the usual q-analogue of n!

Proof.

$$\begin{aligned}
D_{B_n}(q) &= \sum_R q^{d(R)} && (4.4)\\
&= \sum_{\substack{(a_1,a_2,\dots,a_n)\\ a_i\in[0,n-i]}} q^{a_1+a_2+\dots+a_n} && (4.5)\\
&= \left(\sum_{a_1\in[0,n-1]} q^{a_1}\right)\left(\sum_{a_2\in[0,n-2]} q^{a_2}\right)\dots\left(\sum_{a_n\in[0,0]} q^{a_n}\right) && (4.6)\\
&= \left(\sum_{i=0}^{n-1} q^i\right)\left(\sum_{i=0}^{n-2} q^i\right)\dots\left(\sum_{i=0}^{0} q^i\right) && (4.7)
\end{aligned}$$

□

4.2 The Shi Arrangement

The other hyperplane arrangement that we will be discussing is the *Shi arrangement*. The Shi arrangement in $\mathbb{R}^n$ is the set of hyperplanes $x_i - x_j = 0, 1$ for all $1 \leq i < j \leq n$. We denote the by $\mathcal{S}_n$ the Shi arrangement in $\mathbb{R}^n$. In $\mathbb{R}^3$ this arrangement is given in Figure 4.3. Notice that all the intersections between two

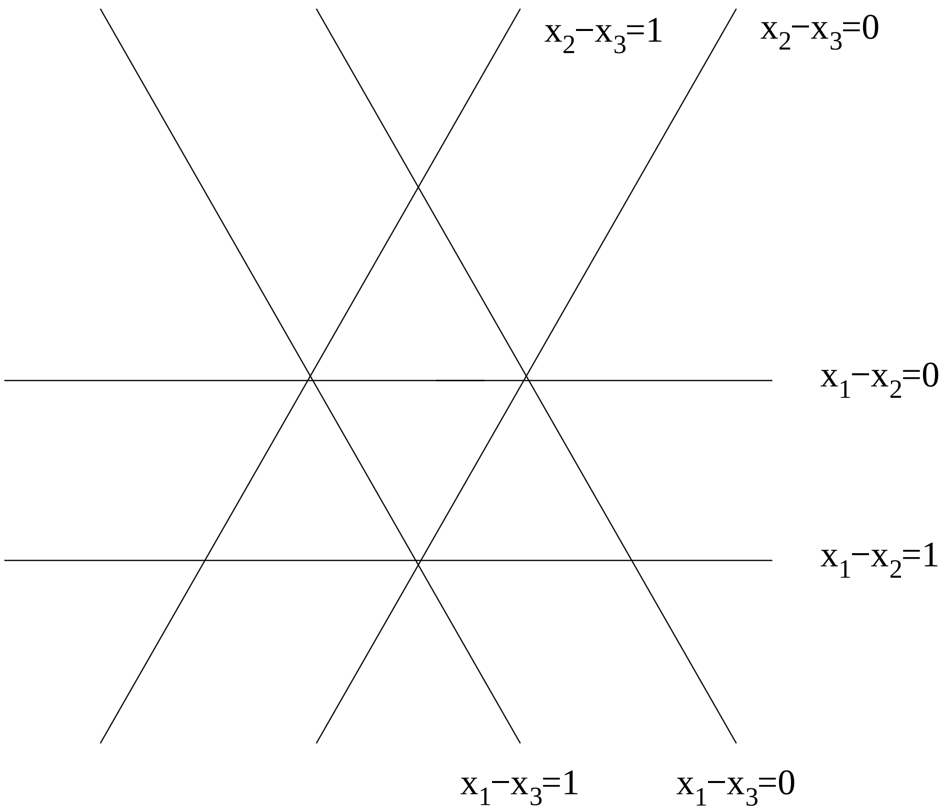

Figure 4.3: The Shi arrangement in $\mathbb{R}^3$ as viewed along the vector $(1, 1, 1)$.

planes are lines parallel to $x_1 = x_2 = x_3$.

Our goal in this and the following two sections is to show that the number of regions in the Shi arrangement in $\mathbb{R}^n$ is $(n+1)^{n-1}$ and to evaluate the distance enumerator for the Shi arrangement. We do the former by presenting a bijection between the regions of the of the Shi arrangement in $\mathbb{R}^n$ and parking functions of length n. We show the latter by presenting a second bijection which allows us to evaluate the distance enumerator. Although the second bijection allows us to evaluate the distance enumerator, the first bijection has the advantage of being relatively simple.

Before we discuss labelling the Shi arrangement, we must find a convenient way to describe the regions of the Shi arrangement. To that end, suppose that R is a region in the Shi arrangement. We notice that the Shi arrangement contains the braid arrangement, hence, specifying a permutation will tell us where R is with respect to the braid arrangement. Suppose the appropriate permutation is ω and, therefore, R lies somewhere in the region $x_{\omega(1)} > x_{\omega(2)} > \ldots > x_{\omega(n)}$. If $i < j$ is *not* an inversion of ω one has that $x_{\omega(i)} > x_{\omega(j)}$ and $\omega(i) < \omega(j)$ which implies that $x_{\omega(i)} - x_{\omega(j)} > 0$. Hence, in these cases, we need to specify more information about where R is, for it may lie in $0 < x_{\omega(i)} - x_{\omega(j)} < 1$ or $x_{\omega(i)} - x_{\omega(j)} > 1$. However, we need not (and, in fact, cannot) specify this for all non-inversion pairs, for some will be implied by transitivity. For example, for the region $x_{\omega(1)} > x_{\omega(2)} > \ldots > x_{\omega(n)} > x_{\omega(1)} - 1$ the fact that $x_{\omega(1)} - x_{\omega(n-1)} < 1$ is implied by the fact that $x_{\omega(n)} > x_{\omega(1)} - 1$ and $x_{\omega(n-1)} > x_{\omega(n)}$. Assuming $\omega(i) < \omega(j) < \omega(k) < \omega(l)$ for $i < j < k < l$, notice that in the region $x_{\omega(1)} > x_{\omega(2)} > \ldots > x_{\omega(n)}$

$$
\begin{aligned}
x_{\omega(j)} - x_{\omega(k)} > 1 \quad &\Longrightarrow \quad x_{\omega(i)} > x_{\omega(j)} > x_{\omega(j)} - 1 > x_{\omega(k)} > x_{\omega(l)} \\
&\Longrightarrow \quad x_{\omega(i)} - 1 > x_{\omega(l)} \\
&\Longrightarrow \quad x_{\omega(i)} - x_{\omega(l)} > 1
\end{aligned}
$$

Hence, we see that if “inner” hyperplane pairs satisfy $x_{\omega(j)} - x_{\omega(k)} > 1$ then so will the “outer” pairs $x_{\omega(i)}$ and $x_{\omega(l)}$, that is $x_{\omega(i)} - x_{\omega(l)} > 1$.

A convenient way to represent the above facts is with a *valid arc arrangement.* Given a region R, to make a valid arc arrangement, one finds the permutation ω that tells us where R is with respect to the contained braid arrangement. Then for any pair i, j such that $x_{\omega(i)} - x_{\omega(j)} > 1$ we draw an arc between $\omega(i)$ and $\omega(j)$. After this is done for all pairs i, j, we remove all arcs that contain other arcs (above, the “inner” planes implication of the “outer” planes). From the above discussion, we see that valid arc arrangements (on some permutation) uniquely specify a region. A valid arc arrangement is always assumed to have an underlying permutation.

Example 4.7 If $\omega = 21378546$ then a valid arc arrangement, $\mathcal{A}$, is in Figure 4.4. Inversions such as $1 < 2$ specify regions such as $x_1 - x_2 < 0$. Of the non-inversions

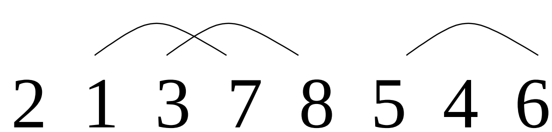

Figure 4.4: A valid arc arrangement.

we have that all of $x_1 - x_7 > 1$, $x_3 - x_8 > 1$, and $x_5 - x_6 > 1$ are true because these pairs happen to be arcs themselves. The non-inversion pair $(2,7)$ contains the pair $(1,7)$ so we must have $x_2 - x_7 > 1$ and the pair $(2,8)$ also contains the pair $(1,7)$ so we also have $x_2 - x_8 > 1$. The rest of the following are all implied (in the same manner as above) by the three arcs $(1,7)$, $(3,8)$ and $(5,6)$; $x_2 - x_5 > 1$, $x_2 - x_4 > 1$, $x_2 - x_6 > 1$, $x_1 - x_8 > 1$, $x_1 - x_5 > 1$, $x_1 - x_4 > 1$, $x_1 - x_6 > 1$, $x_3 - x_5 > 1$, $x_3 - x_4 > 1$ and $x_3 - x_6 > 1$ whereas $x_2 - x_3 < 1$, $x_1 - x_3 < 1$, $x_3 - x_7 < 1$, $x_7 - x_8 < 1$ and $x_4 - x_6 < 1$. $\square$

Theorem 4.8 *The regions of $\mathcal{S}_n$ can be labelled with all the elements of $\mathcal{P}_n$, the parking functions of length n.*

Corollary 4.9 *The number of regions in the Shi arrangement, $\mathcal{S}_n$, is $(n+1)^{n-1}$.*

4.3 The Proof of Theorem 4.8

We now describe our first labelling. Notice that any set of arcs in a valid arc arrangement, partitions $[n]$ into blocks that are chains of increasing integers. For example, the set of arcs in Figure 4.4 gives us the partition $\{\{2\},\{1,7\},\{3,8\},\{5,6\},\{4\}\}$ (of course, chains may have length greater than two). Let $\rho(A)$ be the partition associated with a valid arc arrangement $\mathcal{A}$, and suppose that $\mathcal{A}$ has π as its underlying permutation. Let R be the region in $\mathcal{S}_n$ with valid arc arrangement $\mathcal{A}$. Define a function $\psi_n : \mathcal{S}_n \longrightarrow \mathcal{P}_n$ (of course, this ψ_n has nothing to do with the ψ_n given in Proposition 3.4) such that $\psi_n(R)$ is the sequence $(a_1, a_2, \ldots, a_n)$ where a_i is the position of the smallest member of the block that contains i in $\rho(\mathcal{A})$. Since $a_i \leq \pi^{-1}(i)$ we see that $\psi_n(R)$ is a parking function.

Example 4.10 The parking function we get from the valid arc arrangement $\mathcal{A}$ given in Example 4.7 is $(2, 1, 3, 7, 6, 6, 2, 3)$. □

Proof of Theorem 4.8 (Athanasiadis, Linusson [1, Thm. 2.2]). Proving that ψ_n above is a bijection will prove Theorem 4.8 and that is the route we will take. Given a parking function $(a_1, a_2, \ldots, a_n)$ we build a valid arc arrangement a step at a time. Take all the i such that $a_i = 1$ and put them in a chain (connect them with arcs) in increasing order. Now assume that all the $a_i = j - 1$ have been put into place as a chain with arcs linking the chain in increasing order. Since $(a_1, a_2, \ldots, a_n)$ is a parking function, at least $j - 1$ numbers with arcs have been put into place. Considering all i such that $a_i = j$ put the smallest such i in the j^{th} spot in the sequence and we now note that there is precisely one way to put in the rest of the $a_i = j$ so the resulting string will be a valid arc arrangement i.e. no arc contains another arc. To see why this is true we look a little closer at the structure of a valid arc arrangement, and we prove the claim with an induction argument on the number of terms i such that $a_i = j$ that we have inserted into our growing valid arc arrangement. Since we have placed the first such i (we must place it in the j^{th} position), the base case for our induction is dealt with.

Suppose that i's for which $a_i = j$ are $i_1, i_2, \ldots, i_t$ (it is assumed that $i_1 < i_2 < \cdots < i_t$). Assume that we have placed i_m into our growing valid arc arrangement. Our task is, of course, to place i_{m+1} into our arc arrangement in a unique way. Let $x_1, x_2, \ldots, x_l$ be the numbers to the right of i_m in our arc arrangement (we assume that if $i < j$ then x_i is to the left of x_j). Notice that there are no numbers to the right of i_m in the valid arc arrangement that are the beginning of some chain (in fact, it is clear that there is no number to the right of i_1 that is the beginning of a chain). Since no x_i is the beginning of each chain, the *predecessor* of x_i (the number preceding x_i in the chain containing x_i) is well defined (note that the predecessor of any x_i may be to the left or right of i_m). Let x_p be the first (i.e. p is as small as possible) of the x's such that the predecessor of x_p is to the right of i_m. Suppose the predecessor of x_p is x_k. Hence, in general, our valid arc arrangement must look like the arc arrangement in Figure 4.5. The claim is that we must place i_{m+1} between

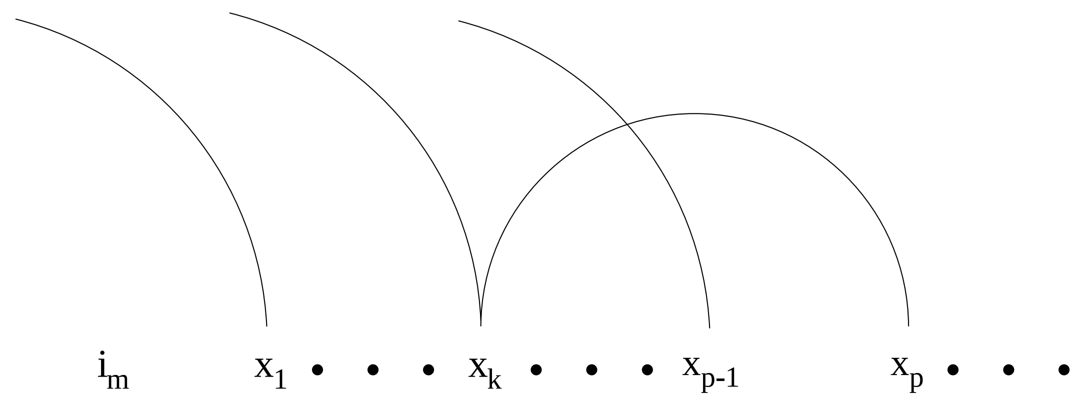

Figure 4.5: What the arc arrangement must look like. The arc emerging from the numbers $x_1, x_2, \ldots, x_{p-1}$ going towards the left indicate that the predecessor of all these numbers lie to the left of i_m.

x_{p-1} and x_p. The reason why no other position for i_{m+1} works is because 1) if i_{m+1} placed before x_{p-1} then the arc from i_m to i_{m+1} will be contained in the arc from the predecessor of x_{p-1} to x_{p-1} and 2) if i_{m+1} is placed after x_p then the arc from i_m to i_{m+1} will contain the arc from x_k and x_p. The reason why placing i_{m+1} in between x_{p-1} and x_p works is because if it did not, then either the arc from i_m to i_{m+1} contains an arc or it is contained in an arc. If the arc from i_m to i_{m+1} contains an arc, our choice of x_p would be contradicted. If, on the other hand, the arc from

i_m to i_{m+1} is contained in an arc then that arc must also contain the arc between x_k and x_p, a contradiction. Thus, there is a unique way to place i_{m+1}, completing our induction.

It is certainly clear that if we apply the above map to a parking function $(a_1, a_2, \ldots, a_n)$ and then apply ψ_n to the result, we get the same parking function with which we started. The converse is also clearly true i.e., if we apply ψ_n to a valid arc arrangement and then apply the above map to the resulting parking function, we will obtain our original arc arrangement. Thus, ψ_n is a bijection. $\square$

The labelling produced by the above for $\mathcal{S}_3$ is given in Figure 4.6.

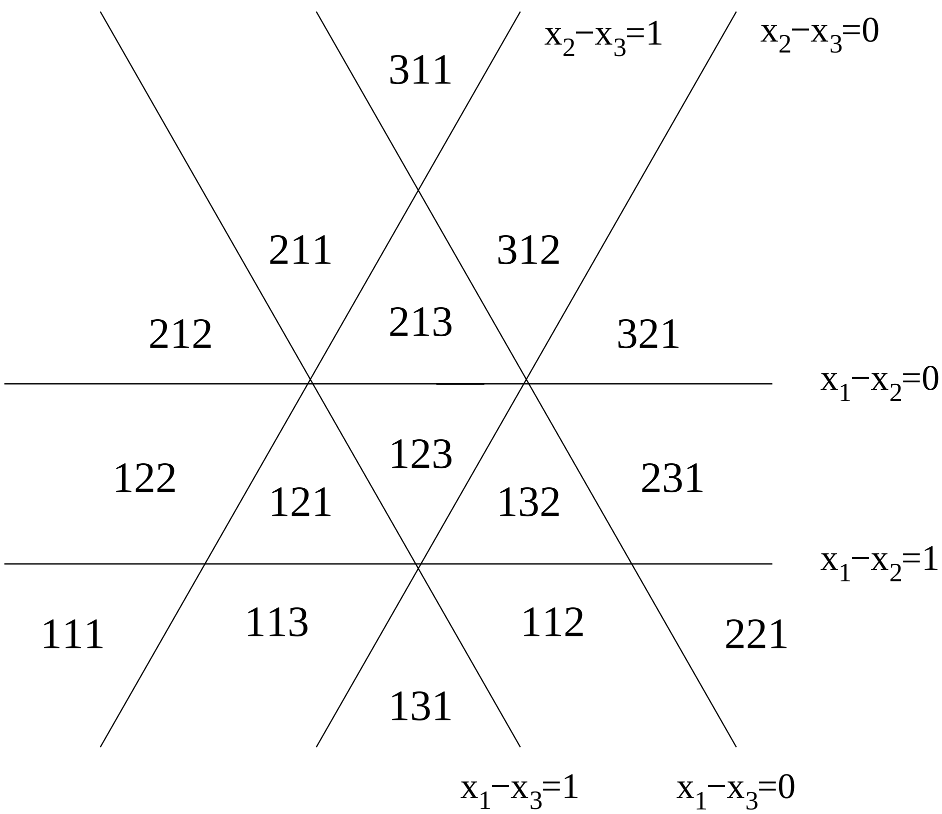

Figure 4.6: The Athanasiadis/Linusson labelling of the Shi arrangement in $\mathbb{R}^3$.

4.4 The Second Bijection between $\mathcal{S}_n$ and $\mathcal{P}_n$

We use a slightly different encoding than a valid arc arrangement to describe a region. A region R in $\mathcal{S}_n$ may be thought of as an ordered pair (ω, I) (which we call a *valid pair*) where $\omega \in \mathfrak{S}_n$ and I is a set of *intervals* $[\omega(i), \omega(j)]$ defined for $i < j$ as $\{\omega(i), \omega(i+1), \ldots, \omega(j)\}$ where (a) $(\omega(i), \omega(j))$ is a non-inversion of the permutation ω and (b) the elements of I form an anti-chain under the relation of set inclusion, i.e. no set of I is a subset of another set of I. The region that is described by a particular (ω, I) is

$$\begin{aligned}
&x_{\omega(1)} > x_{\omega(2)} > \ldots > x_{\omega(n)} \\
&x_{\omega(r)} - x_{\omega(s)} < 1 \text{ if } [\omega(r), \omega(s)] \in I \\
&x_{\omega(r)} - x_{\omega(s)} > 1 \text{ if } r < s, \omega(r) < \omega(s) \text{ and no set} \\
&\qquad\qquad [\omega(i), \omega(j)] \in I \text{ satisfies } i \leq r < s \leq j
\end{aligned}$$

Notice that this is not exactly the same as the description of regions in the last section using the valid arc arrangements, however, they are almost the same. To give a similar description as that given for an arc arrangement we would, for a given region R that lies somewhere in $x_{\omega(1)} > x_{\omega(2)} > \ldots > x_{\omega(n)}$, draw an arc for each $(\omega(i), \omega(j))$ such that $i < j$, $\omega(i) < \omega(j)$ and $x_{\omega(i)} - x_{\omega(j)} < 1$ and remove all the arcs *contained* in another arc (this is where this description differs from the one in the last section, and this difference is caused by the fact that in this case we are drawing arcs for hyperplanes $0 < x_{\omega(i)} - x_{\omega(j)} < 1$ whereas in the last section we drew arcs for the hyperplanes $x_{\omega(i)} - x_{\omega(j)} > 1$). We remove arcs that are contained in other arcs because $i < j < k < l$, $x_{\omega(i)} - x_{\omega(l)} < 1$ and

$$x_{\omega(i)} > x_{\omega(j)} > x_{\omega(k)} > x_{\omega(l)} \tag{4.8}$$

imply that

$$x_{\omega(l)} + 1 > x_{\omega(i)} > x_{\omega(j)} > x_{\omega(k)} > x_{\omega(l)} \tag{4.9}$$

Notice that (4.8) and (4.9) imply that $x_{\omega(r)} - x_{\omega(s)} < 1$ for any $r < s$ and $r, s \in \{i, j, k, l\}$. Thus, we see that arcs inside other arcs are forced.

We allow pairs (ω, I) to be on sets other than $[n]$, namely we allow valid pairs on any t-element subset of $[n]$. In that case, we call (ω, I) a *valid t-pair*. If $i < j$ and $\omega(i) > \omega(j)$ then we call the pair $(\omega(i), \omega(j))$ an *inversion*. Similarly, if $i < j$, $\omega(i) < \omega(j)$ and no interval of I contains both $\omega(i)$ and $\omega(j)$ then we call the pair $(\omega(i), \omega(j))$ *separated*. For any valid pair (ω, I) define

$$\begin{aligned} F(\omega, I, i) &= \{\ j \mid (i, j) \text{ is an inversion }\} \cup \{\ j \mid (i, j) \text{ is separated }\} \\ f(\omega, I, i) &= \#F(\omega, I, i) \end{aligned}$$

Before we describe our labelling, we slightly modify our definition of a parking function. We will maintain the language of Definition 2.4 except that a parking function is a sequence of non-negative integers and require that $b_i \leq i - 1$. Clearly, $(a_1, a_2, \ldots, a_n)$ is a parking function under the old definition if and only if $(a_1 - 1, a_2 - 1, \ldots, a_n - 1)$ is a parking function under the new definition. The reason we do this is so that the weights in the generating function $D_{\mathcal{S}_n}$ work out nicely.

Further, it turns out that this definition is also more convenient for the objects in the next chapter.

Our labelling λ will be as follows. Let R_0 be the region $x_1 > x_2 > \ldots > x_n > x_1 - 1$ and define $\lambda(R_0) = (0, 0, \ldots, 0)$. Now suppose that $\lambda(R)$ has been defined, the only plane separating R and R' is the plane $x_i - x_j = m$ and R and R_0 lie on the same side of $x_i - x_j = m$. In this case, define $\lambda(R')$ to be $\lambda(R) + e_j$ if $m = 0$ and $\lambda(R) + e_i$ if $m = 1$. The labelling for $\mathcal{S}_3$ in this case is in Figure 4.7.

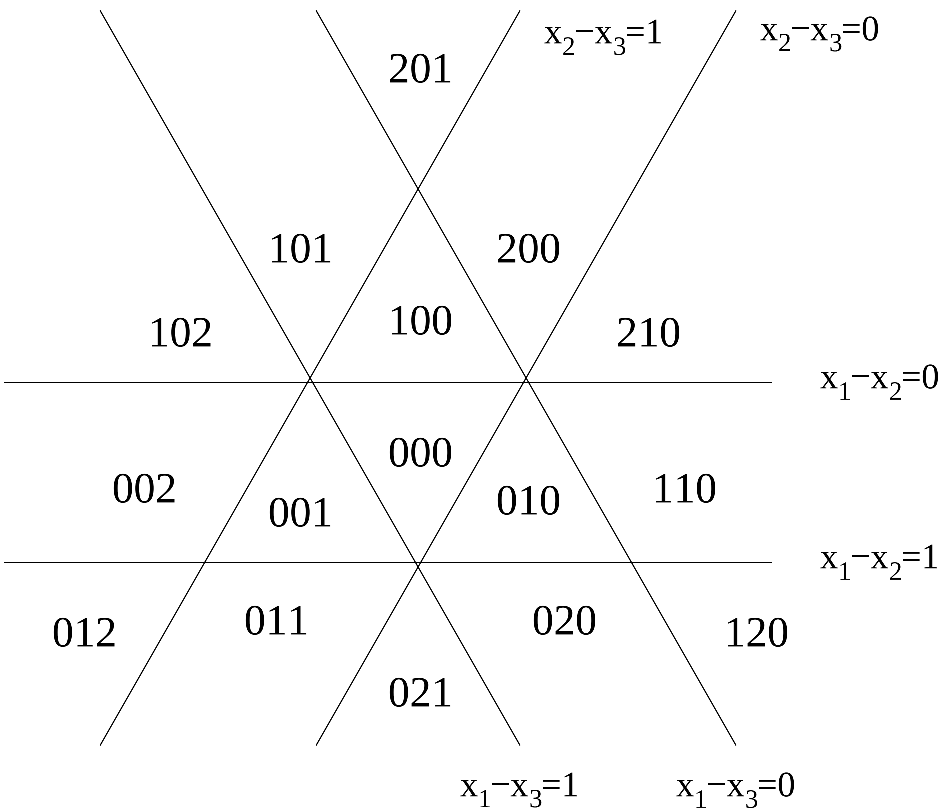

Figure 4.7: The Stanley/Pak labelling λ for $\mathcal{S}_3$.

Notice that the region R_0 is in the middle of all the parallel planes, i.e., the region R_0 satisfies $0 < x_i - x_j < 1$ for all $i < j$. Assuming that R is separated from R_0 by m planes, R is labelled by $a_1, a_2, \ldots, a_n$ and $a_1+a_2+\cdots+a_n = m$ then if R' is labelled as above with $\lambda(R') = \lambda(R)+e_i$ then $m+1$ planes separate R' from R_0 and the sum of the entries in $\lambda(R')$ is $= a_1+a_2+\ldots+a_{i-1}+a_i+1+a_{i+1}+\ldots+a_n = m+1$. Thus, we see that for any region R, if R is labelled $a_1, a_2, \ldots, a_n$ then

$$d(R) = a_1 + a_2 + \cdots + a_n \tag{4.10}$$

Furthermore, from the above argument we can see that $a_{\omega(i)}$ is the number of planes such that

1. $x_{\omega(j)} - x_{\omega(i)} = 0$ separates R from R_0 where $(\omega(i), \omega(j))$ is an inversion
2. $x_{\omega(i)} - x_{\omega(j)} = 1$ separates R from R_0 where $(\omega(i), \omega(j))$ is separated

i.e. $a_{\omega(i)} = f(\omega, I, \omega(i))$. Hence,

$$\lambda(R) = (f(\omega, I, 1), f(\omega, I, 2), \ldots, f(\omega, I, n))$$

It is clear that $\lambda(R)$ is a parking function because $a_{\omega(i)} = f(\omega, I, \omega(i))$ cannot be greater than $n - \omega(i)$ the number of elements to the right of $\omega(i)$ in ω.

Example 4.11 For the valid pair (ω, I) where $\omega = 71342865$ and $I = \{[1,4],[3,8],[4,6]\}$ we get the parking function $\alpha = (4,1,3,2,0,1,7,2)$. Indeed, $F(\omega, I, 7) = \{1,3,4,2,6,5\} \cup \{8\}$ which implies that $f(\omega, I, 7) = 7$. □

The Second Bijection between $\mathcal{S}_n$ and $\mathcal{P}_n$. (Stanley, Pak [25, Thm. 2.1]). Given a parking function $(a_1, a_2, \ldots, a_n)$ we build a valid (ω, I) step by step. We show that there will be at most one way to build (ω, I), implying that the map $R \mapsto \lambda(R)$ is injective. From the Athanasiadis/Linusson proof of Theorem 4.8, we know that the number of regions in the Shi arrangement is $(n+1)^{n-1}$ allowing us to conclude that the map $R \mapsto \lambda(R)$ is a bijection.

After the m^{th} step we will have a valid m-pair (ω^m, I^m) and we will build it up with the condition that $f(\omega^m, I^m, i) = a_i$ for all i in ω^m. Let $b_1 b_2 \ldots b_n$ be the permutation obtained from $a_1, a_2, \ldots, a_n$ by listing the indices i such that $a_i = 0$ from greatest to least, the indices i such that $a_i = 1$ from greatest to least and so on. For example, the parking function $\alpha = (4,1,3,2,0,1,7,2)$ above would have $b_1 b_2 \ldots b_n = 56248317$. We obtain ω^m by inserting b_m into the permutation ω^{m-1} and adding an appropriate interval $[b_m, c_m]$ to I^{m-1} and remove intervals from I^{m-1} so that I^m remains an anti-chain. We describe this precisely now.

Because at all stages we require $f(\omega^m, I^m, i) = a_i$ for every $i \in \omega^m$, we must have that $f(\omega^{m-1}, I^{m-1}, h) = f(\omega^m, I^m, h)$ for all h in ω^{m-1}. This means that when we insert b_m into ω^{m-1} we cannot insert it to the right of a larger term c (because then we would create another inversion with c and, therefore, $f(\omega^m, I^m, c) > f(\omega^{m-1}, I^{m-1}, c)$) and we cannot insert it to the right of a smaller term c unless there exists a $d > c$ such that d would be to the right of b_m and (c, d) is not separated (for otherwise we would create another separated pair with c and, again, we would have $f(\omega^m, I^m, c) > f(\omega^{m-1}, I^{m-1}, c)$). Furthermore, we cannot choose $[b_m, c_m]$ to include any separate pairs in (ω^{m-1}, I^{m-1}) because we want separated pairs to remain separated. This is because if $i < j$ and were separated in (ω^{m-1}, I^{m-1}) and not in (ω^m, I^m) then we would have $f(\omega^{m-1}, I^{m-1}, i) > f(\omega^m, I^m, i)$. The claim is that with the above stipulations, there is only one way to insert b_m and choose the interval $[b_m, c_m]$. We show this claim now.

First note that once we have inserted b_m there is only one way to choose the interval $[b_m, c_m]$ given that $f(\omega^m, I^m, b_m) = a_{b_m}$ i.e. if b_m is inserted after $\omega^{m-1}(p)$ then c_m is chosen as the integer such that $c_m > b_m$ and the number of terms after c_m is $a_{b_m} - j$ where j is the number of inversions (b_m, i). Thus, let us assume that there are two places that we can insert b_m, say after $\omega^{m-1}(p)$ and $\omega^{m-1}(j)$ where $p < j$, and that the resulting permutations are ω^m and $\bar{\omega}^m$, respectively. Hence, a portion of ω^m and $\bar{\omega}^m$ look like $\ldots, p, b_m, \ldots, j, \ldots$ and $\ldots, p, \ldots, j, b_m, \ldots$, respectively. Further, let $[b_m, c_m]$ and $[b_m, d_m]$ be the intervals added to I^{m-1} to obtain I^m and $\bar{I}^m$ respectively. Since we want $f(\omega^{m-1}, I^{m-1}, h) = f(\omega^m, I^m, h)$, we know that $\omega^{m-1}(j) < b_m$. Clearly, then we must have that $c_m < d_m$ (since $f(\omega^m, I^m, b_m) = f(\bar{\omega}^m, \bar{I}, b_m)$ and the fact that $(b_m, \omega^m(j))$ is an inversion in ω^m but not in $\bar{\omega}^m$) which implies that $(\omega^m(j), d_m)$ is separated in (ω^m, I^m) (there must be at least one less separation in (ω^m, I^m)). Hence, $[\omega^{m-1}(j), d_m]$ is separated in (ω^{m-1}, I^{m-1}). But because separated pairs must remain separated, $(\omega^{m-1}(j), d_m)$ is separated in $(\bar{\omega}^m, \bar{I}^m)$, a contradiction. Thus, there is at most one way to choose (ω^m, I^m) for each m and, thus, at most one choice for (ω, I). $\square$

Corollary 4.12

$$D_{\mathcal{S}_n}(q) = \sum_{(a_1, a_2, \ldots, a_n)} q^{a_1 + a_2 + \cdots + a_n} \tag{4.11}$$

where the sum is over all parking functions $(a_1, a_2, \ldots, a_n) \in \mathcal{P}_n$.

Notes and References

§4.1 Most of the material in this section comes from two papers by R. Stanley. The material on the braid arrangement can be found in Stanley [25, Intro.][22]. The author's treatment of the braid arrangement in the former paper is much more thorough than the latter, however the latter gives a quick, concise introduction to the material. A nice concise look at permutation inversions and other elementary permutation statistics is given in Stanley [20, Sec. 1.3]. In particular, the Proposition 4.4 can be found there.

§4.2 - §4.4 Concerning the Shi arrangement, this hyperplane arrangement was first considered in Shi [16][17] and later in Headley [8]. The elementary proof of Theorem 4.8 is given in Athansiadis and Linusson [1, Thm. 2.2]. Further, in [1] the authors give the convenient labelling, of the Shi arrangement, we called valid arc arrangements. The second bijection between $\mathcal{S}_n$ and $\mathcal{P}_n$ (given in Section 4.4) can be found in Stanley [25, Thm. 2.1]. In [25], Stanley states the theorem in the earlier paper [22, Thm. 5.1], where it is stated without proof. Although the proof is omitted, this paper expediently deals with some interesting topics concerning hyperplane arrangements, trees, interval orders and parking functions. In [22], Stanley collaborated with I. Pak on the proof given here.

Chapter 5

Tree Inversions

Inversions in trees are a weight on trees that have an interesting connection with parking functions, as described in this chapter. We assume that a tree on $n+1$ vertices will have the vertex labels $[n]' = \{0, 1, \ldots, n\}$. We denote a rooted tree by (T, r) where r is the root of T.

Definition 5.1 *Given a rooted tree (T, r) a tree inversion is an ordered pair (i, j) where $i < j$ and the unique path connecting i to the root contains the vertex j. Define $\mathrm{inv}(T, r)$ to be the number of inversions of (T, r).*

Example 5.2 Consider the tree T in Figure 5.1 with root 0. The inversions in the tree are the ordered pairs $(11, 13), (4, 7), (1, 7)$ and $(2, 5)$ and, therefore, $\mathrm{inv}(T, 0) = 4$. $\square$

We note that if the root r is not 0 then any vertex with a label i smaller than the root will create the inversion (i, r). If no root is specified, it will be assumed that

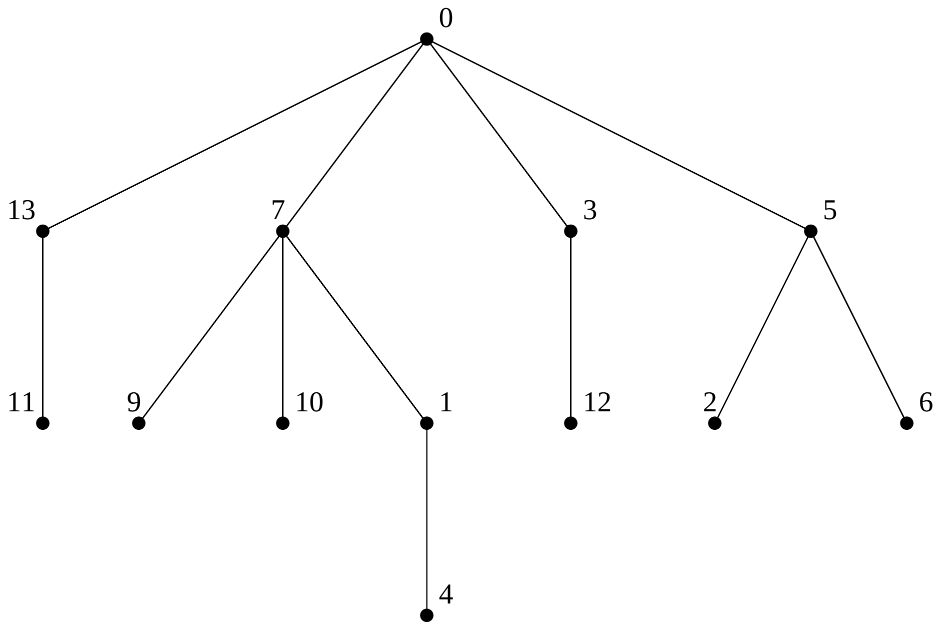

Figure 5.1: A tree with root 0.

the root of the tree is the vertex labelled 0, i.e. $(T, 0)$ will be written as T. In most cases, the root of our tree will be 0.

5.1 The Parking Function Generating Function and the Inversion Enumerator for Trees

We define two generating functions that will make the connection between parking functions and tree inversions explicit. We use the modified definition of a parking function that we used for the second proof of Theorem 4.8 in the last chapter; that is, a sequence of non-negative integers $(a_1, a_2, \ldots, a_n)$ is a parking function if its non-decreasing rearrangement $(b_1, b_2, \ldots, b_n)$ satisfies $b_i \leq i - 1$. We will soon see that this is a convenient modification in this context. We now define a generating function pertaining to parking functions. First, we define the weight ω_n of a parking function to be

$$\omega_n(a_1, a_2, \ldots, a_n) = \binom{n}{2} - \sum_{i=1}^{n} a_i \tag{5.1}$$

To see what this weight function means geometrically, we note that

$$\begin{aligned} \omega_n(a_1, a_2, \ldots, a_n) &= \binom{n}{2} - \sum_{i=1} a_i \\ &= \sum_{i=1} i - \sum_{i=1} a_i \\ &= \sum_{i=1} (i - a_i) \end{aligned}$$

and, hence we see that the weight of a parking function is the area between the unit step function (beginning at 0) and the parking function, when the non-decreasing rearrangement of the parking function is displayed as a lattice path from 0 to n.

Example 5.3 For the parking function $p = (1, 0, 0, 3, 5, 7, 3, 3)$ displayed in Figure 5.2, we see its weight is 6. □

Define the *parking function generating function* to be

$$P_n(q) = \sum_{(a_1, a_2, \ldots, a_n)} q^{\omega_n(a_1, a_2, \ldots, a_n)} \tag{5.2}$$

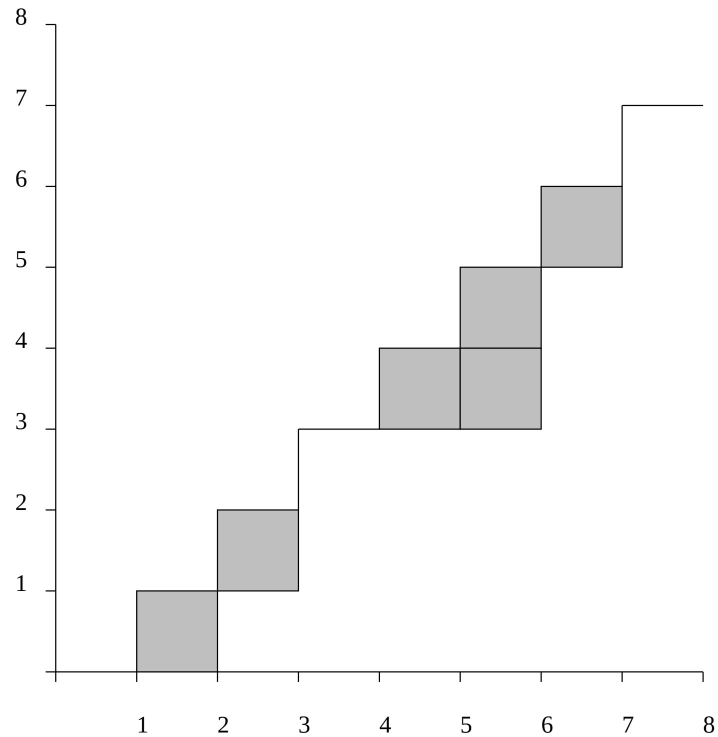

Figure 5.2: A graphical look at the weight of $(1,0,0,3,5,7,3,3) \in \mathcal{P}_8$.

where the above sum is over all parking functions of length n. We define, for trees, the *inversion enumerator* to be

$$I_n(q) = \sum_{T\,:\,|V(T)|=n+1} q^{\mathrm{inv}(T)} \tag{5.3}$$

The generating function $I_n(q)$ for the first few values of n is

$I_1(q) = 1$

$I_2(q) = 1$

$I_3(q) = 2 + q$

$I_4(q) = 6 + 6q + 3q^2 + q^3$

$I_5(q) = 24 + 36q + 30q^2 + 20q^3 + 10q^4 + 4q^5 + q^6$

Example 5.4 In the case of $n = 3$ the three trees are displayed in Figure 5.3. The

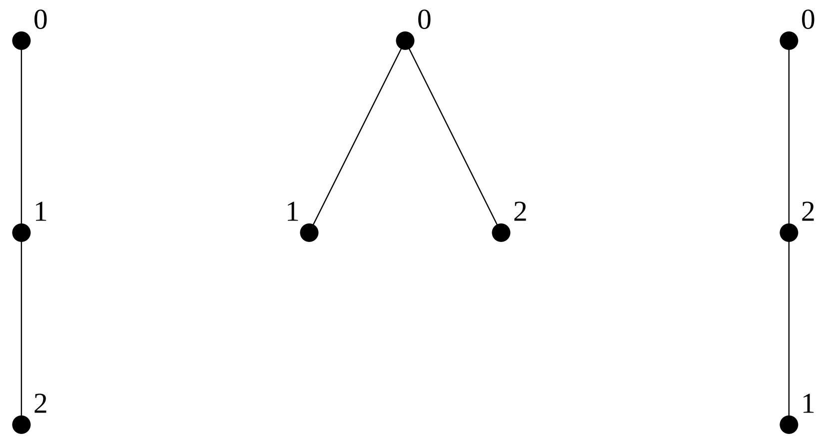

Figure 5.3: The trees on three vertices.

first two trees in Figure 5.3 are the trees with no inversions and the last tree is the tree with one inversion, giving $I_2(q) = 2 + q$. □

The above generating functions have some very remarkable properties which we will now present as a theorem. Parts (a) and (c) of the following theorem can be found in Kreweras [10, Thm. I] whereas part (b) can be found in Stanley [25, Thm. 3.1].

Theorem 5.5 (a) $I_n(q)$ *satisfies*

$$I_n(1 + q) = \sum_G q^{e(G)-n} \tag{5.4}$$

where G ranges over all connected graphs (without loops or multiple edges) on $n + 1$ labelled vertices, and where $e(G)$ denotes the number of edges of G.

(b) *We have the following two generating function identities*

$$\sum_{n\geq 0} I_n(q)(q-1)^n \frac{x^{n+1}}{(n+1)!} = \log \sum_{n\geq 0} q^{\binom{n}{2}} \frac{x^n}{n!} \tag{5.5}$$

$$\sum_{n\geq 0} I_n(q)(q-1)^n \frac{x^n}{n!} = \frac{\sum_{n\geq 0} q^{\binom{n+1}{2}} \frac{x^n}{n!}}{\sum_{n\geq 0} q^{\binom{n}{2}} \frac{x^n}{n!}} \tag{5.6}$$

(c)

$$I_n(q) = P_n(q) \tag{5.7}$$

Before we prove the above theorem, we discuss some of its consequences. From (a) and (c) of Theorem 5.5 we get the expansion

$$P_n(1+q) = \sum_{G} q^{e(G)-n} \tag{5.8}$$

(where G has the same range as in part (a)) giving us a nice result concerning parking functions and connected graphs. An interesting consequence of part (c) is that the number of trees with no inversions is $n!$ (since the number of trees with no inversions correspond to parking functions of lowest weight, i.e., parking functions whose non-decreasing rearrangement is $1, 2, \ldots, n$ and there are $n!$ of these parking functions). Further, since the number of trees on $n+1$ vertices is $(n+1)^{n-1}$, it follows from part (c) that $P_n(1) = (n+1)^{n-1}$, giving another proof of Proposition 2.10. As well, from (5.8) we see that $P_n(2)$ is the number of connected graphs on

$n+1$ vertices. We prove Thm 5.5 below, but before we do this we must introduce a useful concept and prove a lemma.

Let G be a connected graph on the vertex set $[n]'$. We define a certain spanning tree τ_G on G with the following rule. Start at the vertex 0 and when at any vertex move to the adjacent vertex with the largest label, if possible, otherwise backtrack.

Example 5.6 For the graph G in Figure 5.4, τ_G is given in Figure 5.5. The order that the vertices were traversed in applying the above algorithm to obtain τ_G is $0, 5, 4, 3, 4, 2$ and 1. □

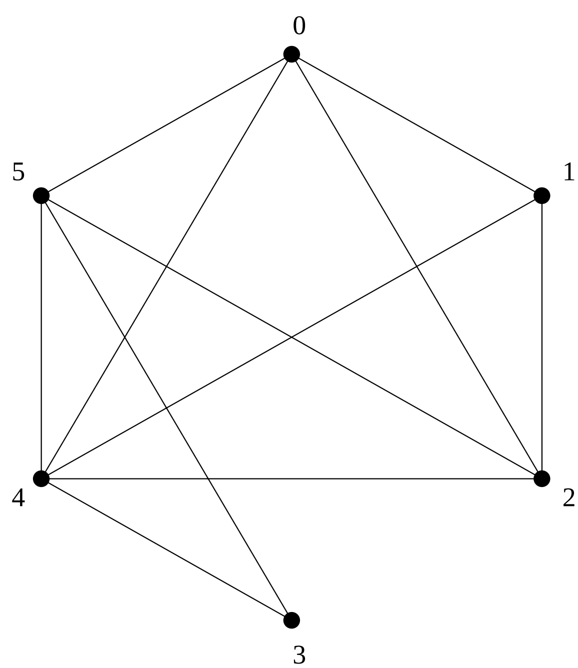

Figure 5.4: A connected graph G.

The following lemma can be found in Stanley [24, Ex. 5.48].

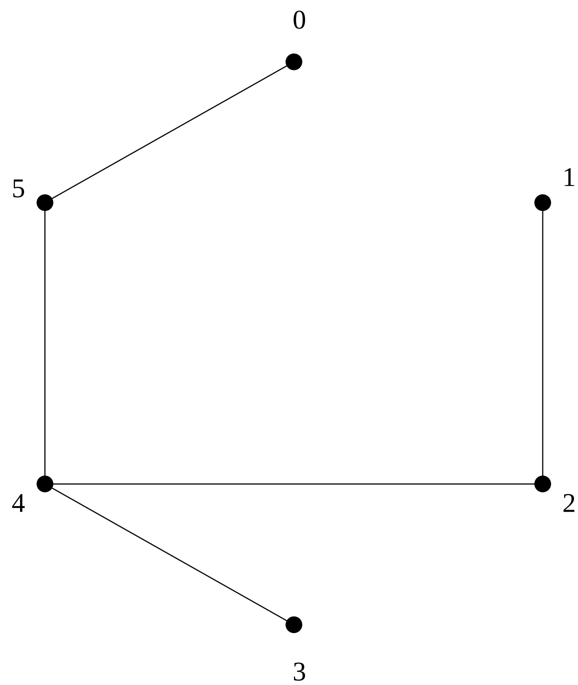

Figure 5.5: τ_G for the graph in Figure 5.4.

Lemma 5.7 *Let τ be a tree on $[n]'$. Then for any connected graph G on $[n]'$, we have $\tau_G = \tau$ if and only if every edge of G that is not in τ has the form $\{i, k\}$, where (i, j) is an inversion of τ and k is the parent of j in τ.*

Proof. Suppose that τ is a tree on $[n]'$ and G is a connected graph on the same vertex set. Further, suppose that every edge of G not in τ has the form given in the lemma. For a contradiction, assume that $\tau_G \neq \tau$. In this case, there exists an edge $e = \{i, k\}$ of τ_G that is not in τ. Since e is an edge of G not in τ, k has a child j such that (i, j) is an inversion in τ. We may assume that the edge e has the property that the distance from k to 0 is minimized. Hence the path connecting k to 0 coincides for τ_G and τ. It is clear from the above, when the algorithm we use to obtain τ_G was run, it must at some point arrive at k via this path. Because $\{i, k\}$ is an edge of τ_G, we must choose i next in our algorithm. But $j > i$, and hence we should choose j instead of i at this point, unless j has already been chosen in the algorithm. But in that case, the algorithm must have chosen j, backtracked and then visited k. But this is impossible, since we wouldn't have backtracked from the vertex j, we would have picked k (either right away, or after picking some other vertices and then backtracking to j), a contradiction.

Conversely, suppose that $\tau_G = \tau$ and, again for a contradiction, that there exists an edge e of G not in τ such that $e = \{i, k\}$ and the child j of k on the path from k to i in τ is such that (i, j) is NOT an inversion. This is clearly impossible because when we ran the algorithm to get τ_G at some point we chose the vertex k and since j and i are both descendants of k in τ, they both must be chosen after k. Hence, they would both be available for choosing and since (i, j) is not an inversion, $i > j$ and we would have chosen $\{i, k\}$ to be in our tree, a contradiction. □

Notice that the number of inversions in a tree does not depend on the actual labels on a tree, it just depends on the relative ordering of the labels. Thus, in general, we can have a tree with $n + 1$ vertices whose labels come from any subset of $\mathbb{N}$ of size $n + 1$. Given any subset $A \subseteq \mathbb{N}$ such that $|A| = n + 1$ and any $i \in A$ define the *rank of* i (denoted $\mathrm{rank}(i)$) to be 1 plus the number of elements in A less than i. Further notice that given two sets V and V', both subsets of $\mathbb{N}$, there is a map from trees on the first vertex set to trees on the second vertex set that preserves the number of inversions and is given by mapping the vertex of rank i in V to the vertex of rank i in V'. We will be using this relabelling repeatedly in what follows.

Now for the proof of Theorem 5.5.

Proof of Theorem 5.5. (a) For any tree τ on $[n]'$ we define the set $\imath(\tau)$ to be the set of *inversion edges* defined in Lemma 5.7, i.e. the set of edges $\{i,k\}$ such that the child j of k on the path in τ from i to k have the property that (i,j) is an inversion of τ. Now, if we fix a τ and consider all connected graphs G on $[n]'$ with $\tau_G = \tau$, we see that any such G is simply τ together with some subset of $\imath(\tau)$. Therefore,

$$\sum_G q^{e(G)-n} = (1+q)^{\mathrm{inv}(\tau)}$$

where the sum is over all connected graphs G on $n+1$ vertices such that $\tau_G = \tau$. Further, Lemma 5.7 states that all connected graphs G can be obtained uniquely via the above method, i.e. by adding the edges of some subset of $\imath(\tau_G)$ to τ_G. Hence, summing over all trees τ we get

$$\sum_G q^{e(G)-n} = \sum_\tau (1+q)^{\mathrm{inv}(\tau)} \tag{5.9}$$

where the first sum is now over all connected graphs G with $n+1$ vertices. Noting that the right hand side of (5.9) is $I_n(1+q)$, we have our result.

(b) The first of the equations, (5.5), is a direct result of the exponential formula (see Stanley [24, Cor. 5.1.6]). To see this, clearly $(1+q)^{\binom{n}{2}}$ is the generating function for a graph on n vertices where q^i marks a graph with i edges. Hence,

$$\exp\left(\sum_{n\geq 1}\sum_{G} q^{e(G)}\frac{x^n}{n!}\right) = \sum_{n\geq 0}(1+q)^{\binom{n}{2}}\frac{x^n}{n!}$$

where the summation over G is over all connected graphs G on n vertices and from (a) the left hand side becomes

$$\exp\left(\sum_{n\geq 0} I_n(1+q)q^n\frac{x^{n+1}}{(n+1)!}\right) = \sum_{n\geq 0}(1+q)^{\binom{n}{2}}\frac{x^n}{n!}$$

giving us

$$\sum_{n\geq 0} I_n(q)(q-1)^n\frac{x^{n+1}}{(n+1)!} = \log\left(\sum_{n\geq 0} q^{\binom{n}{2}}\frac{x^n}{n!}\right) \tag{5.10}$$

We note that (5.6) follows by applying $\frac{d}{dx}$ to (5.10).

(c) We prove this by showing that the two generating functions in question satisfy the same recursion, viz they both satisfy

$$F_{n+1}(q) = \sum_{i=0}^{n}\binom{n}{i}(q^i + q^{i-1} + \ldots + 1)F_i(q)F_{n-i}(q) \tag{5.11}$$

with $F_0(q) = 1$. We will begin with the parking function generating function.

Given a parking function $p = (a_1, a_2, \ldots, a_{n+1})$, from Proposition 2.7 we can decompose p into two parking functions $p' = (a'_1, a'_2, \ldots, a'_l)$ and $p'' = (a''_1, a''_2, \ldots, a''_{n-l+1})$, a subset of size l of $[n]$ (the set A_1 given in Proposition 2.7) and an integer $a^*_{n+1} - a_{n+1} \in [l]'$ (also given in Proposition 2.7). Conversely, given two parking functions $p' = (a'_1, a'_2, \ldots, a'_l)$ and $p'' = (a''_1, a''_2, \ldots, a''_{n-l+1})$ of length l and $n-l+1$ respectively, a subset A_1 of size l of $[n]$ and an integer $a^*_{n+1} - a_{n+1} \in [l]'$ we can construct a parking function p by the following: For each $i \in A_1$ define a_i to be a'_j where i is the j^{th} smallest element of A_1. Similarly, for each $i \in [n]\backslash A_1$ define a_i to be $a''_j + l$ where i is the j^{th} smallest element of $[n]\backslash A_1$. Further,

$$\begin{aligned}
\omega_l(p') + \omega_{n-l+1}(p'') + a^*_{n+1} - a_{n+1} &= \binom{l}{2} - \sum_{i=1}^{l} a'_i + \binom{n+1-l}{2} - \sum_{i=1}^{n+1-l} a''_i \\
&\quad + a^*_{n+1} - a_{n+1} \\
&= \sum_{i=1}^{l-1} i - \sum_{i \in A_1} a_i + \sum_{i=l+1}^{n} i - \sum_{i \in A_2} (a''_i + l) \\
&\quad + a^*_{n+1} - a_{n+1} \\
&= \sum_{i=1}^{n} i - \sum_{i=1}^{n} a_i - a_{n+1} \\
&= \omega_{n+1}(p)
\end{aligned}$$

where the second last line follows from the fact that $a^*_{n+1} = l$ (see Proposition 2.7). Hence, there is a bijection between parking functions p of length $n+1$ and quadruples $(p', p'', A_1, a^*_{n+1} - a_{n+1})$ such that the weight of $\omega_{n+1}(p) = \omega_l(p') + \omega_{n-1+1}(p'') + a^*_{n+1} - a_{n+1}$ (where p corresponds to the quadruple $(p', p'', A_1, a^*_{n+1} - a_{n+1})$). Thus, we have

$$P_{n+1}(q) = \sum_{i=0}^{n} \binom{n}{i} (q^i + q^{i-1} + \ldots + 1) P_i(q) P_{n-i}(q)$$

and since, clearly, $P_0(q) = 1$, we see that $P_n(q)$ satisfies (5.11).

Now, we prove that $I_n(q)$ satisfies (5.11) by proving that a tree can be similarly decomposed. Clearly, $I_0(q) = 1$. Given a tree T on $[n+1]'$, let e be the edge on the path from 0 to $n+1$ that is incident with 0. Removing e from the graph we obtain two trees T_1 and (T_2, r) on the vertex sets V_1 and V_2 respectively, $|V_2| = i+1$ and where r is the other vertex incident with e on the tree T. We note that $V_2 \subseteq [n+1]$ and necessarily $n+1 \in V_2$. Notice that $\mathrm{inv}(T) = \mathrm{inv}(T_1) + \mathrm{inv}(T_2, r)$. If we replace the root r of T_2 with a vertex labelled 0, and make it the new root, then notice that we lose r' inversions, where r' is $\mathrm{rank}(r) - 1$. Call this new tree T_2'. We can relabel the vertices of T_1 and T_2' with the vertex sets $[n-i]'$ and $[i]'$, respectively, preserving the order of the labels and, therefore, preserving the number of inversions in the two trees. Thus, we obtain a quadruple (T_1, T_2', V, r') where T_1 is a tree on $[n-i]'$, T_2' is a tree on $[i]'$, a set $V = V_2 \backslash \{n+1\}$ and a number r' between 0 and i. Notice that $\mathrm{inv}(T) = \mathrm{inv}(T_1') + \mathrm{inv}(T_2') + r'$. Conversely, suppose that we are given a quadruple (T_1, T_2, V, r'), where T_1 is a tree on the $[n-i]'$, T_2 is a tree on the $[i]'$, V is a subset of $[n]$ of size i and r' is a number in $[i]'$, then we can construct a tree T with $\mathrm{inv}(T) = \mathrm{inv}(T_1) + \mathrm{inv}(T_2) + r'$ in the following way: First, we change the non-zero labels in T_1 to labels in $[n] \backslash V$, maintaining the relative order of the vertices to obtain a new tree T_1'. Second, we obtain a new tree, T_2', from the tree T_2 by replacing the non-zero labels on T_2 with $V \cup \{n+1\} \backslash \{r\}$ where r is the vertex of rank $r'+1$, in preserving their order. Next we relabel the root 0 with the label r. Now attach the vertex labelled 0 of T_1' to the new root, r, of T_2' to get T. Notice that $\mathrm{inv}(T) = \mathrm{inv}(T_1') + \mathrm{inv}(T_2') + r'$. Summing over all i will account for all trees and hence we see that

$$I_{n+1}(q) = \sum_{i=0}^{n} \binom{n}{i} (q^i + q^{i-1} + \ldots q + 1) I_i(q) I_{n-i}(q),$$

so $I_n(q)$ satisfies (5.11). $\square$

Notice the above proof gives us an recursive bijection between parking functions of weight m and trees with m inversions. To make this more convincing, since both parking functions and tree inversions satisfy (5.11) we have the following correspondence

$$p \longleftrightarrow (A, p', p'', r) \longleftrightarrow (V, T_1, T_2, r) \longleftrightarrow T$$

where

1. p is a parking function of length $n+1$
2. (A, p', p'', r) is a quadruple where
   (a) A is some i-subset of $[n+1]$
   (b) p' is a parking function with labels in A
   (c) p'' is a parking function on the labels $[n+1] \backslash A$
   (d) r is a number between 0 and i

and

1. T is a tree on $[n+1]'$
2. (V, T_1, T_2, r) is a quadruple where

(a) V is an i-subset of $[n]$

(b) T_1 is a tree on the vertices $n+1-i$ vertices $[n]'\backslash V$

(c) T_2 is a tree on the $i+1$ vertices $V \cup \{n+1\}$

(d) r is a number between 0 and i

The r in the parking function case is obtained from the difference $a^*_{n+1} - a_{n+1}$ whereas the r in the case of trees tells us which vertex of T_2 should be made the root of T_2. This fact will be exploited in the next chapter.

The following corollary gives a connection between the inversion enumerator for trees and the distance enumerator for the Shi arrangement.

Corollary 5.8

$$I_n(q) = q^{\binom{n}{2}} D_{S_n}(1/q).$$

Proof. This follows from Corollary 4.12 and Theorem 5.5.

Notes and References

§5.1. The proof of Theorem 5.5(a) and Theorem 5.5(b) can be found in Stanley [24, Ex. 5.48]. In it, most of the proof is left as an exercise and Stanley gives the full reference as follows: (5.6) was first proved in Mallows and Riordan [13] using an indirect generating function method. The bijection given in Theorem 5.5(a) can be found in Gessel and Wang [6] in which the authors give a simple, yet useful, combinatorial bijection. The proof used in Theorem 5.5(c) can be found in

Kreweras [10, Thm. I]. In this paper, Kreweras begins with (5.11) and describes some objects that satisfy the recursion.

With regard to tree inversions, we note that there are bijections from parking functions to trees that don't preserve the statistic tree inversion. For example in Foata and Riordan [5, Sec. 2], they give a mapping (which the authors credit to H. Pollak) that maps parking functions to Prüfer codes. They describe this map "... [the map] is simplicity itself.", which it is. They also give another mapping in Section 3 of that paper by putting both parking functions and trees into one-to-one correspondence with another set.

Chapter 6

Generalizations of Parking Functions

We now consider two generalizations of parking functions. Initially, the two generalizations seem very similar but there will be subtle and important differences between the two. The first one we will call $\hat{k}$-parking functions and the second k-parking functions. It turns out that we can, likewise, generalize the objects in the previous chapters and the generalizations fit well with at least one of $\hat{k}$-parking functions or k-parking functions. We will see that k-parking functions are much more complex than $\hat{k}$-parking functions.

6.1 $\hat{k}$-Parking Functions

A *$\hat{k}$-parking function of length n* is a sequence of positive integers $(a_1, a_2, \ldots, a_n)$ such that its non-decreasing rearrangement $(b_1, b_2, \ldots, b_n)$ satisfies $b_i \leq ki$. Notice that the number of $\hat{k}$-parking functions is $k^n(n+1)^{n-1}$. This is true because any $\hat{k}$-parking function $(a_1, a_2, \ldots, a_n)$ can be written as the sum $k(q_1, q_2, \ldots, q_n) - (r_1, r_2, \ldots, r_n)$ where $(q_1, q_2, \ldots, q_n)$ is a $\hat{1}$-parking function (notice that $\hat{1}$-parking functions are simply parking functions) and $0 \leq r_i \leq k-1$. Denote the set of $\hat{k}$-parking functions by $\hat{\mathcal{P}}_n^{(k)}$.

6.1.1 Noncrossing Partitions

The first object that we generalize from the previous chapters are noncrossing partitions. A *k-noncrossing partition of the set* $\{1, 2, \ldots, kn\}$ is a noncrossing partition of the set $\{1, 2, \ldots, kn\}$ such that every block has size divisible by k. Denote by $\mathrm{NC}_{n+1}^{(k)}$ the set of k-noncrossing partitions. We can consider $\mathrm{NC}_{n+1}^{(k)}$ as a poset in the same way that we considered NC_{n+1} a poset. Notice, however, that $\mathrm{NC}_{n+1}^{(k)}$ does not have a $\hat{0}$. Even so, it is clear that every maximal chain in $\mathrm{NC}_{n+1}^{(k)}$ has the same length, namely n. Suppose that π and σ are k-noncrossing partitions such that $\pi < \sigma$ and such that no other k-noncrossing partition τ satisfies $\pi < \tau < \sigma$. Clearly, σ must be obtained from π by merging two blocks B and B' of π. Suppose that $\min B < \min B'$. Define

$$\Lambda(\pi, \sigma) = \max\{\, i \mid i \in B \text{ and } i < \min B' \} \tag{6.1}$$

and for the chain $\mathtt{m} = (\pi_0, \pi_1, \ldots, \pi_n)$ in $\mathrm{NC}_{n+1}^{(k)}$ define

$$\Lambda(\mathtt{m}) = (\Lambda(\pi_0, \pi_1), \Lambda(\pi_1, \pi_2), \ldots, \Lambda(\pi_{n-1}, \pi_n))$$

We prove that Λ is a bijection between the maximal chains of $\mathrm{NC}_{n+1}^{(k)}$ and $\hat{\mathcal{P}}_n^{(k)}$.

Example 6.1 In $\mathrm{NC}_{n+1}^{(1)}$, let the following be the maximal chain $\mathtt{m}$

$$
\begin{gathered}
1-2-3-4-5 \\
\downarrow \\
13-2-4-5 \\
\downarrow \\
135-2-4 \\
\downarrow \\
1345-2 \\
\downarrow \\
12345
\end{gathered}
$$

where $\pi_0 = 1-2-3-4-5$ and so on. We see that $\Lambda(\pi_0, \pi_1) = 1$, $\Lambda(\pi_1, \pi_2) = 3$, $\Lambda(\pi_2, \pi_3) = 3$ and $\Lambda(\pi_3, \pi_4) = 1$ implying that $\Lambda(\mathtt{m}) = (1, 3, 3, 1)$. $\square$

Before we give a proof that Λ is a bijection, we first make a note about parking functions. Notice that the number of i's occurring in a parking function is at most $n-i+1$. Further, any sequence $(a_1, a_2, \ldots, a_n)$ that has at most $n-i+1$ occurrences of i's, for all $i = 1, 2, \ldots, n$ is a parking function. The necessity of the above condition follows from Definition 2.4 and the sufficiency follows from Definition 2.5. This, clearly, carries over into $\hat{k}$-parking functions, namely that a sequence $(a_1, a_2, \ldots, a_n)$ is a $\hat{k}$-parking function if and only if the number of a_j satisfying $k(i-1)+1 \leq a_j \leq ki$ is at most $n-i+1$. Now for the proof, which can be found in Stanley [23, Thm. 3.1].

Theorem 6.2 *Λ is a bijection between maximal chains of $\mathrm{NC}_{n+1}^{(k)}$ and $\hat{\mathcal{P}}_n^{(k)}$.*

Proof. Let $\mathtt{m} = (\pi_0, \pi_1, \ldots, \pi_n)$ be a maximal chain in $\mathrm{NC}_{n+1}^{(k)}$. Notice that if $\Lambda(\mathtt{m}) = (a_1, a_2, \ldots, a_n)$ then $a_s = r$, for some $k(i-1)+1 \leq r \leq ki$, $1 \leq i \leq n$ if and only if π_s is obtained from π_{s-1} by joining a block B and B' where $r \in B$ and 1) $r = \max B$ and $r < \min B'$ *or* 2) there exists an $l \in B$ such that $l > \max B'$. In either case, all of the elements of B' are greater than $k(i-1)+1$. Clearly, there can be at most $n-i+1$ blocks with this property. Further, every time we have an a_s such that $k(i-1)+1 \leq a_s \leq ki$, there is one less block in π_s than in π_{s-1} that has the property that every one of its members is greater than $k(i-1)+1$. Hence, there are at most $n-i+1$ members of $(a_1, a_2, \ldots, a_n)$ that lie in between $k(i-1)$ and ki. Thus, $(a_1, a_2, \ldots, a_n)$ is a $\hat{k}$-parking function.

Next we show that Λ is injective. To that end, suppose that $\Lambda(\mathtt{m}) = \Lambda(\mathtt{m}') = (a_1, a_2, \ldots, a_n)$. Let $r = \max\{\ a_i \mid 1 \leq i \leq n\}$ and $s = \max\{\ i \mid a_i = r\}$. The first claim is that there exists a block B in π_{s-1} such that $B = \{r+1, r+2, \ldots, k+r\}$. We prove the claim now.

If r and $r+1$ are in the same block of π_{s-1} then it is impossible that $a_s = r$. Hence, r and $r+1$ are in different blocks in π_{s-1} and assume that $r \in B$ and $r+1 \in B'$. If there exists an $i \in B'$ such that $i < r$, then by the noncrossing property of π_s, we must combine B and B' to get π_s (otherwise, if we combined B with some other block B'' then the blocks $B \cup B''$ would "cross" the block B' in π_s). However, then $a_s < r$ (because then $a_s \in B'$ and, hence, could not be r). Therefore, B' does not contain elements less than r. If B' has more than k elements in it then for some $t < s$, $a_t > r$ (because we would have combined the block containing $r+1$

with another block containing elements all greater than r to make B'). Finally, if $B' = \{r+1, r+2, \ldots, j-1, j+ki+1, \ldots\}$ (i.e. B' is not a block of the form $\{m, m+1, \ldots, m+k\}$) then for some $t > s$, $a_t = j-1 > r$ (this follows from the noncrossing property of $\pi_0, \pi_1, \ldots, \pi_n$. Thus, the block B' containing $r+1$ is $\{r+1, r+2, \ldots, k+r\}$.

Our next claim is that π_s is obtained by combining the block that contains r, B, and the block that contains $r+1$, B'. If not, then in order for a_s to equal r we must merge B with a block B'' all of whose elements are greater than r. At some point $t > s$ we must merge the block containing r, $B^{(3)}$, and the block containing $r+1$, $B^{(4)}$. Since $B \cup B' \subseteq B^{(3)}$, $B^{(3)}$ contains r and elements greater than all the elements in B' (the noncrossing property of π_{t-1}). Hence, when we merge $B^{(3)}$ with $B^{(4)}$ to obtain π_t we get $a_t = r$, contradicting our choice of s.

It is clear now how we can uniquely recover $\mathtt{m}$ from the $\hat{k}$-parking function $p = (a_1, a_2, \ldots, a_n)$ by induction. Namely, remove a_s from p, and call the new sequence p', and notice that p' is still a $\hat{k}$-parking function. By induction, there exists a maximal chain $\mathtt{m}' = (\pi_0, \pi_1, \ldots, \pi_{s-1}, \pi_{s+1}, \ldots, \pi_n)$ in $\mathrm{NC}_n^{(k)}$. In all the π_i, replace every number j greater than r in every block with $j+k$. For each π_i such that $i \leq s-1$ add the block $\{r+1, r+2, \ldots, k+r\}$ and call these new blocks $\pi'_0, \pi'_1, \ldots, \pi'_{s-1}$. For each π_i such that $i \geq s+1$ take the block containing r and form its union with $\{r+1, r+2, \ldots, k+r\}$ and call these new block $\pi'_{s+1}, \pi'_{s+2}, \ldots, \pi'_n$. Finally, define π'_s the same as π_{s-1} except that we merge the block containing r with the block $\{r+1, r+2, \ldots, k+r\}$.

Notice that the above proof suffices to show that Λ is surjective, since the last paragraph holds for an arbitrary $\hat{k}$-parking function. $\square$

Corollary 6.3 *The number of maximal chains in* $\mathrm{NC}_{n+1}^{(k)}$ *is* $k^n(n+1)^{n-1}$.

6.1.2 Tree Inversions

A *rooted $\hat{k}$-tree* (T, r) on $n+1$ vertices is a rooted tree on $n+1$ vertices such that each edge is assigned a colour $0, 1, \ldots, k-1$. Notice that if we fix the root (to be 0 in most cases) then the number of rooted $\hat{k}$-trees is clearly $k^n(n+1)^{n-1}$. In a tree T, let $\kappa(e)$ be the colour of the edge e and define $\mathrm{Path}(v, w)$ to be the set of edges on the unique path from v to w. Define inv_k to be

$$\mathrm{inv}_k(T, r) = \mathrm{inv}(T, r) + \sum_{v} \sum_{e \in \mathrm{Path}(v,r)} \kappa(e)$$

where $\mathrm{inv}(T, r)$ is the number of ordinary inversions in a tree (as defined in Definition 5.1). We define the *inversion enumerator for $\hat{k}$-trees* as

$$I_n^{(k)}(q) = \sum_{T\,:\,|V(T)|=n+1} q^{\mathrm{inv}_k(T)} \tag{6.2}$$

where the sum is over all rooted $\hat{k}$-trees.

Example 6.4 In Figure 6.1 we have the rooted k-tree T with root 0. Here we have $k = 5$ and the number of vertices is 8. Since, $\mathrm{inv}(T) = 3$ and

$$\sum_{v} \sum_{e \in \mathrm{Path}(v,r)} \kappa(e) = 26$$

we have $\mathrm{inv}_k(T) = 29$. □

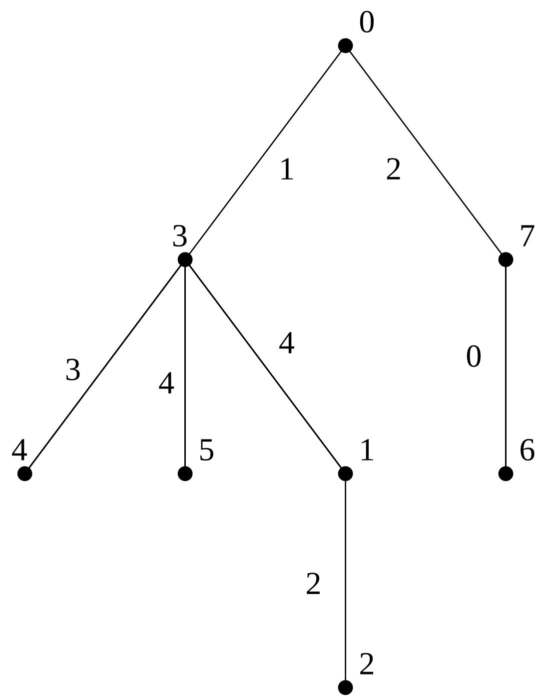

Figure 6.1: A rooted $\hat{k}$-tree T rooted at 0.

Proposition 6.5 *There exists a bijection $\hat{\theta}_n$ between the set of $\hat{k}$-parking functions of length n and rooted $\hat{k}$-trees that preserves the weights in the generating functions $\hat{P}_n^{(k)}(q)$ and $\hat{I}_n^{(k)}(q)$, i.e. $\hat{\theta}_n$ implies $\hat{P}_n^{(k)}(q) = \hat{I}_n^{(k)}(q)$.*

Proof. The proof is almost identical to the proof of Theorem 5.5(c) except for the following fact. In that proof, the difference $a_n^* - a_n$ gives one the information of what root to give the tree T_2, as it was called. In the case with $\hat{k}$-parking functions the difference $a_m^* - a_m$ can be at most kj. Thus, we see that this difference gives the information of what vertex is the root of T_2 but it also tells us the colour of the edge connecting the root of T_2 with the vertex labelled 0 of T_1. This information about the colour of the edge connecting the two roots is obtained by dividing $a_n^* - a_n$ by j to obtain $tj + r$ where t is the quotient and r is the remainder. We then colour the edge connecting the two roots t, adding tj to the weight of the tree (since there are j vertices in T_2). □

We note that Thm 5.5(c) relied heavily upon Proposition 2.7. This does not present a problem because a similar proposition holds for $\hat{k}$-parking functions. The statement of this proposition can be found below in Lemma 6.7 except we must replace all occurences of $k(j-1)$ in the lemma with kj.

6.2 k-Parking Functions

Definition 6.6 *A k-parking function of length n is a sequence $(a_1, a_2, \dots, a_n)$ of non-negative integers such that its non-decreasing rearrangement $(b_1, b_2, \dots, b_n)n$ satisfies $b_i \leq k(i-1)$. We denote by $\mathcal{P}_n^{(k)}$ the set of all k-parking functions of length n.*

Lemma 6.7 *Let $(a_1, a_2, \dots, a_n)$ be a k-parking function of length n. Let a_n^* be the largest number such that $(a_1, a_2, \dots, a_{n-1}, a_n^*)$ is a k-parking function. Then a_n^* is equal to $k(j-1)$ for some j. Furthermore, a_n^* is the only term in $(a_1, a_2, \dots, a_{n-1}, a_n^*)$ that is equal to $k(j-1)$ and in the non-decreasing reordering $(b_1, b_2, \dots, b_n)$ of $(a_1, a_2, \dots, a_{n-1}, a_n^*)$, we have $b_j = a_n^*$.*

Proof. The proof for this is completely analogous to the proof of Proposition 2.6 and 2.7. □

For the k-parking function $(a_1, a_2, \ldots, a_n)$ we call the k-parking function $(a_1, a_2, \ldots, a_{n-1}, a_n^*)$ the *reduced complement* of $(a_1, a_2, \ldots, a_n)$. In our next results we will count the number of k-parking functions of length n. However, we will first prove a stronger result we will be using later. Define the *k-parking function generating function* as

$$P_n^{(k)}(q) = \sum_{(a_1,a_2,\ldots,a_n)\in\mathcal{P}_n^{(k)}} q^{\omega_n^{(k)}(a_1,a_2,\ldots,a_n)} \tag{6.3}$$

where

$$\omega_n^{(k)}(a_1, a_2, \ldots, a_n) = k\binom{n}{2} - \sum_{i=1}^{n} a_i$$

We find a recursion for $P_n^{(k)}(q)$ and use it to compute the number of k-parking functions of length n. The following theorem is due to Yan [27, Thm. 7].

Theorem 6.8

$$P_{n+1}^{(k)}(q) = \sum_{j=0}^{n} \binom{n}{j}(1 + q + \ldots + q^{kj})(1 + q + \cdots + q^{k-1})^{n-j} P_j^{(k)}(q) P_{n-j}^{(1)}(q^k)$$

for $n \geq 1$ and $P_0^{(k)}(q) = 1$.

Proof. We denote by $\mathcal{P}_{n,j}^{(k)}$ the set of k-parking functions of length n whose reduced complement has a_n^* equalling kj. Clearly, if the reduced complement of $(a_1, a_2, \ldots, a_n)$ is $(a_1, a_2, \ldots, a_{n-1}, kj)$ then any other k-parking function $(a_1, a_2, \ldots, a_{n-1}, x)$, where $x \leq kj$ also has reduced complement $(a_1, a_2, \ldots, a_{n-1}, kj)$.

Thus,

$$P_{n+1}^{(k)}(q) = \sum_{j=0}^{n}(1 + q + \cdots + q^{kj})P_{n+1,j}^{(k)}(q) \tag{6.4}$$

Fixing a k-parking function p in $\mathcal{P}_{n+1,j}^{(k)}$, consider the two sets A_1 and A_2 defined by

$$A_1 = \{\ i \mid a_i < kj\}$$

and

$$A_2 = \{\ i \mid a_i > kj\}$$

From Lemma 6.7 we see that the union of these two sets has cardinality n. Let

$$p' = (a_i)_{i \in A_1}$$

and

$$p'' = (a_i - kj)_{i \in A_2}$$

It is clear that p' can be any arbitrary k-parking function of length j. Notice that p'' is a sequence of length $n - j$ and can be written as

$$p'' = k(q_1, q_2, \ldots, q_{n-j}) + (r_1, r_2, \ldots, r_{n-j})$$

where $(q_1, q_2, \ldots, q_{n-j})$ can be any arbitrary 1-parking function of length $n-j$ and $(r_1, r_2, \ldots, r_{n-j})$ is a $n-j$-tuple with $0 \le r_i \le k-1$, i.e. p'' can be any $\hat{k}$-parking function. Hence, the weight of p'' is,

$$k\binom{n-j+1}{2} - \sum_{i \in A_2}(a_i - kj)$$

Thus, we see that (noting that we have $\binom{n}{j}$ choices for the set A_1)

$$P^{(k)}_{n+1,j}(q) = \binom{n}{j}(1+q+\cdots+q^{k-1})^{n-j}P^{(k)}_j(q)P^{(1)}_{n-j}(q^k) \tag{6.5}$$

Substituting (6.5) into (6.4)

$$P^{(k)}_{n+1}(q) = \sum_{j=0}^{n}\binom{n}{j}(1+q+\cdots+q^{kj})(1+q+\cdots+q^{k-1})^{n-j}P^{(k)}_j(q)P^{(1)}_{n-j}(q^k) \tag{6.6}$$

□

Proposition 6.9 $P^{(k)}_n(1) = (kn+1)^{n-1}$*, i.e. the number of k-parking functions of length* n *is* $(kn+1)^{n-1}$ *for* $n \ge 0$ *and* $k \ge 1$.

Proof. Denote by $p^{(k)}_n$ and $p^{(k)}_{n,j}$ the number of elements in $\mathcal{P}^{(k)}_n$ and the number of elements in $\mathcal{P}^{(k)}_{n,j}$, respectively. Clearly, if we take $P^{(k)}_{n+1}(q)$ and substitute in $q=1$ we will get the number of k-parking functions of length $n+1$. If this process is carried out on (6.6) we get

$$p^{(k)}_{n+1} = \sum_{j=0}^{n}\binom{n}{j}(kj+1)k^{n-j}p^{(k)}_j p^{(1)}_{n-j}$$

Multiplying both sides by $\frac{x^n}{n!}$ and summing over all $n \geq 0$ we get

$$\frac{d}{dx}P^{(k)}(x) = \left(kx\frac{d}{dx}P^{(k)}(x) + P^{(k)}(x)\right)H(x) \tag{6.7}$$

where

$$P^{(k)}(x) = \sum_{n\geq 0} p_n^{(k)}\frac{x^n}{n!}$$

and

$$H(x) = \sum_{n\geq 0} k^n(n+1)^{n-1}\frac{x^n}{n!}$$

From (6.7) we obtain the differential equation

$$\frac{d}{dx}P^{(k)}(x) = \frac{H(x)}{1-kxH(x)}P^{(k)}(x)$$

and since $P^{(k)}(0) = p_0^{(k)} = 1$, we divide by $P^{(k)}(x)$ to get

$$\frac{d}{dx}\log(P^{(k)}(x)) = \frac{H(x)}{1-kxH(x)} \tag{6.8}$$

Applying the Lagrange Inversion Formula (see Goulden and Jackson [7, Sec. 1.2] Stanley [24, Sec. 5.4]) to

$$y = x\exp(ky) \tag{6.9}$$

and

$$\exp(ky)$$

one finds

$$H(x) = \exp(ky) \tag{6.10}$$

Now, applying $x\frac{d}{dx}$ to (6.9) and solving for $x\frac{dy}{dx}$, we obtain

$$x\frac{dy}{dx} = \frac{y}{1-ky}$$

Substituting (6.9) and (6.10) into the right hand side of the above gives us

$$x\frac{dy}{dx} = \frac{x\exp(ky)}{1-kx\exp(ky)} = \frac{xH(x)}{1-kxH(x)}$$

so (6.8) becomes

$$\frac{d}{dx}\log(P^{(k)}(x)) = \frac{dy}{dx}$$

and we conclude that

$$\log(P^{(k)}(x)) = y$$

since $y(0) = 0$ from the definition of y. Thus,

$$P^{(k)}(x) = \exp(y)$$

and applying Lagrange's Theorem, again, gives

$$\begin{aligned} p_n^{(k)} &= n![x^n]\exp(y) \\ &= \frac{n!}{n}[x^{n-1}]\exp(kx)^n \exp(x) \\ &= (kn+1)^{n-1} \end{aligned}$$

for $n \geq 1$ and $p_0^{(k)} = 1$. □

6.2.1 Hyperplane Arrangements

We say very little about hyperplane arrangements here in this chapter except that the proof given in Section 4.4 can be generalized to k-parking functions. The generalization of the Shi arrangement, called the *extended Shi arrangement* and denoted by $\mathcal{S}_n^k$, is the collection of hyperplanes

$$x_i - x_j = -k+1, -k+2, \ldots, k, \text{ for } 1 \leq i < j \leq n$$

The proof is similar to that given in Section 4.4 except this time the regions are described by a $k+1$-tuple $(\omega, I_1, I_2, \ldots, I_k)$ where the permutation ω has the same function as in the case $k = 1$ and I_m specifies which coordinates are a distance m from each other. Of course, as in the case of $k = 1$ there are certain compatibility requirements between ω and the I_m's that must be satisfied. As for the case $k = 1$, given a k-parking function $(a_1, a_2, \dots, a_n)$, we build up the $k+1$-tuple $(\omega, I_1, I_2, \dots, I_k)$ one step at a time.

A consequence of this is that the distance enumerator for the extended Shi arrangement $\mathcal{S}_n^k$ is given by

$$D_{\mathcal{S}_n^k}(q) = \sum_{\substack{(a_1,a_2,\cdots,a_n) \\ (a_1,a_2,\cdots,a_n)\in\mathcal{P}_n^{(k)}}} q^{a_1+a_2+\cdots+a_n}$$

6.2.2 Tree Inversions

In this section we give a result that generalizes most of Theorem 5.5. We will be generalizing the notions of tree inversions and connected rooted graphs.

A *rooted k-tree* on $n+1$ vertices is a rooted tree T with root r on $n+1$ vertices whose non-root edges (root edges are edges that emerge from the root) can be coloured one of the k colours $0, 1, \dots, k-1$. Root edges are coloured with the colour 0. We denote a rooted k-tree by (T, r). If no root is specified we assume that the root is the vertex labelled 0. We define the *inversion enumerator for rooted k-trees*, denoted by $I_n^{(k)}(q)$, the same as $\hat{I}_n^{(k)}(q)$ except that we sum over all rooted k-trees instead of rooted $\hat{k}$-trees (of course). Another object that we need is a *connected k-graph*. A connected k-graph (G, r) on the $n+1$ vertices $[n]'$ is a graph G that has its edges coloured any one of the k colours $0, 1, \dots, k-1$. Here, r is a distinguished vertex, which we will call the *root* and r has the property that any edges emerging from it must have colour 0. We don't allow loops but two edges can have the same endpoint as long as they don't have the same colour. Define the generating function $C_n^{(k)}(q)$ to be

$$C_n^{(k)}(q) = \sum_{G\,:\,G \text{ connected}} q^{e(G)-n} \tag{6.11}$$

where $e(G)$ is the number of edges of G.

Example 6.10 In Figure 6.2 we have a rooted connected k-graph G with root 0. Here, $k = 4$ and the number of vertices is 6. The numbers beside an edge indicate that the edge is possibly a multiple edge with the colours indicated. Since there are 10 edges, we see that $e(G) - n = 10 - 5 = 5$. □

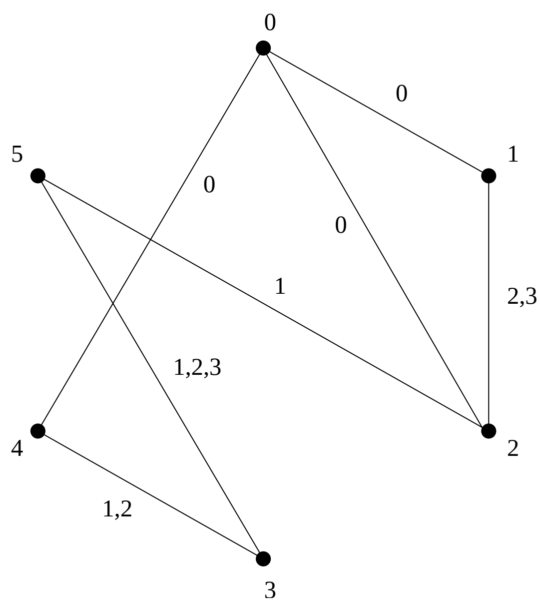

Figure 6.2: A rooted graph G with root 0. The numbers beside each edge indicate that the edge is, in fact, a multiple edge with the colours listed as the colours of the edges.

Without delay we prove the generalization of the Theorem 5.5. The proof is by showing that $P_n^{(k)}(1+q)$ and $C_n^{(k)}(q)$ satisfy the same recursion and $I_n^{(k)}(1+q)$ and

$C_n^{(k)}(q)$ satisfy the same recursion. The following theorem can be found in Stanley [25, Thm. 3.3] and Yan [27].

Theorem 6.11 *$P_n^{(k)}(1+q) = C_n^{(k)}(q) = I_n^{(k)}(1+q)$ for $n \geq 0$ and $k \geq 0$.*

Proof. It was shown in Theorem 6.8 that the generating function $P_n^{(k)}(q)$ satisfies the recurrence

$$P_{n+1}^{(k)}(q) = \sum_{j=0}^{n} \binom{n}{j} (1+q+\cdots+q^{k-1})^{n-j}(1+q+\cdots+q^{kj})P_j^{(k)}(q)P_{n-j}^{(1)}(q^k)$$

and hence

$$P_{n+1}^{(k)}(1+q) = \sum_{j=0}^{n} \binom{n}{j} \left(\frac{(1+q)^k-1}{q}\right)^{n-j} \frac{(1+q)^{kj+1}-1}{q} P_j^{(k)}(1+q)P_{n-j}^{(1)}((1+q)^k) \tag{6.12}$$

We show that $C_{n+1}^{(k)}(q)$ also satisfies the recursion given in (6.12).

Given a connected k-graph G on the $n+2$ vertices $[n+1]'$, (in which we assume the root to be 0) we delete the vertex labelled 1. This will split G up into two sets of components, one set consisting of the component containing the vertex 0 and the other set consisting of the components not containing the vertex labelled 0. Consider the subgraph consisting of the components not containing the vertex 0, the vertex 1 and all the edges from 1 to the components not containing 0. Call this subgraph K and suppose that it has $i+1$ vertices. Let L be $G \backslash K$ (see Figure 6.3 to see the subgraphs K and L in an arbitrary graph). Clearly, both K and L

are connected. Hence we have

$$C_{n+1}^{(k)} = \sum_{K \cup L} q^{e(L)-(n-i)} \cdot q^{e(K)-i+1} \cdot q^{d(G)}$$

where $d(G)$ is the number of edges from the vertex 1 to the subgraph L. There are

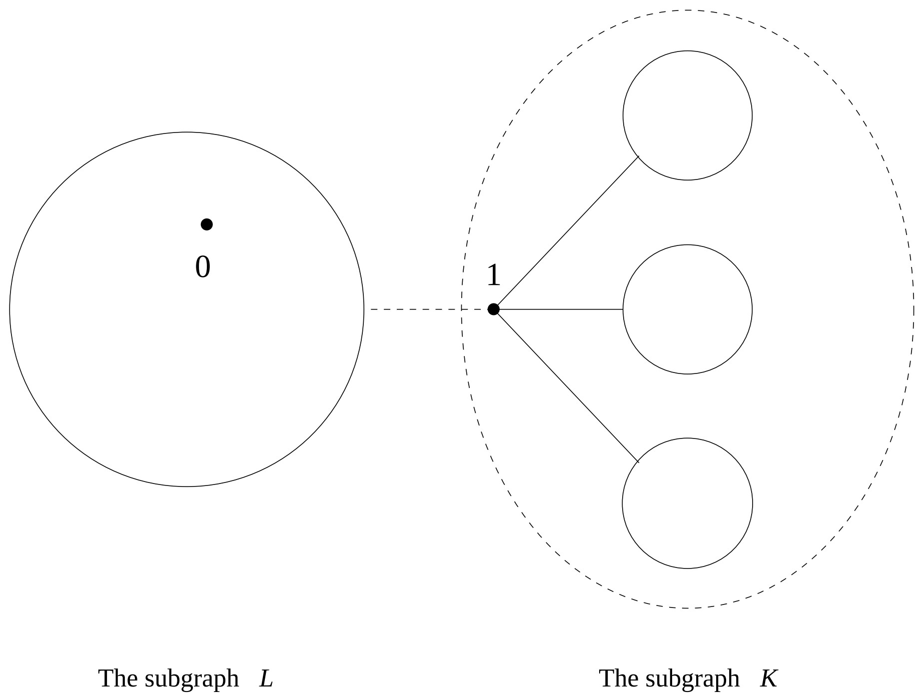

Figure 6.3: The subgraphs K and L. Of course, the "edges" between 1 and components merely indicate that 1 is adjacent to a component and not the number of edges between 1 and that component.

$\binom{n}{i}$ ways to choose the vertices in K (since the vertex 1 must be in K). Notice that K does not have the structure of a connected k-graph, since what we may consider as the root (the vertex 1) can have edges emerging from it that are not coloured 0. In fact, since K has no distinguished vertex in this way, its structure is similar to that of a connected 1-graph. However, between any two vertices of K there can be any subset of edges coloured $0, 1, \ldots, k-1$. Hence, we see that

$$q^{-(i+1)} \sum_{K} q^{e(G)} = \sum_{K' : K' \text{ a connected 1-graph}} \left(q^{-(n+1)} \sum_{j=1}^{k} \binom{k}{j} q^j \right)^{e(K')} \tag{6.13}$$

$$= q^{-(i+1)} \Big(((1+q)^k - 1)^i \cdot C_i^{(1)}((1+q)^k - 1) \Big) \tag{6.14}$$

$$= q^{-1} \left(\frac{(1+q)^k - 1)}{q} \right)^i \cdot C_i^{(1)}((1+q)^k - 1) \tag{6.15}$$

Clearly, the subgraph L is a just a connected k-graph on $n-i$ vertices and, hence,

$$\sum_{L} q^{e(L)-(n-i)} = C_{n-i}^{(k)}(q)$$

The possible values of $d(G)$ can be computed as follows. If 0 and 1 are neighbours then 1 can have i neighbours in the rest of L, where i is between 0 and $k(n-i)$ (remembering that we can have edges with the same endpoints as long as they have different colours). Or, if 0 and 1 are not neighbours then there must be at least one edge from 1 to the rest of L and there can be as many as $k(n-i)$. Hence,

$$\begin{aligned}
C_{n+1}^{(k)}(q) &= \sum_{G} q^{e(G)-n} \\
&= \sum_{K\cup L} q^{e(K)-(i+1)} \cdot q^{e(L)-(n-i)} \cdot q^{d(G)} \\
&= \sum_{i=0} \binom{n}{i} C_{n-i}^{(k)}(q) q^{-1} \left(\frac{(1+q)^k-1}{q}\right)^i C_i^{(1)}((1+q)^k-1) \times \\
&\quad \left(q \sum_{j=0}^{k(n-i)} \binom{k(n-i)}{j} q^j + \sum_{j=1}^{k(n-i)} \binom{k(n-i)}{j} q^j\right) \\
&= \sum_{i=0} \binom{n}{i} q^{-1} \left(\frac{(1+q)^k-1}{q}\right)^i C_{n-i}^{(k)}(q) C_i^{(1)}((1+q)^k-1) \times \\
&\quad \left((q+1) \sum_{j=0}^{k(n-i)} \binom{k(n-i)}{j} q^j - 1\right) \\
&= \sum_{i=0} \binom{n}{i} \left(\frac{(1+q)^k-1}{q}\right)^i \frac{(1+q)^{k(n-i)+1}-1}{q} \times \\
&\quad C_{n-i}^{(k)}(q) C_i^{(1)}((1+q)^k-1)
\end{aligned}$$

This completes the first equality. To prove the second we will show that $I_n^{(k)}(q)$ satisfies the recursion

$$\begin{aligned}
I_n^{(k)}(q) &= \sum_{a_1+a_2+\cdots+a_{k+1}=n} \binom{n}{a_1, a_2, \ldots, a_{k+1}} \cdot q^{a_2+2a_3+\cdots+(k-1)a_k} \times \\
&\quad (1+q+\cdots+q^{a_1+a_2+\cdots+a_k} I_{a_1}^{(k)} I_{a_2}^{(k)} \cdots I_{a_{k+1}}^{(k)} \qquad (6.16)
\end{aligned}$$

and $C_n^{(k)}(q)$ satisfies the recursion

$$C_{n+1}^{(k)}(q) = \sum_{a_1+a_2+\cdots+a_{k+1}=n} \binom{n}{a_1, a_2, \ldots, a_{k+1}} \cdot (1+q)^{a_2+2a_3+\cdots+(k-1)a_k} \times \frac{(1+q)^{a_1+a_2+\cdots+a_{k+1}+1}-1}{q} C_{a_1}^{(k)} C_{a_2}^{(k)} \cdots C_{a_{k+1}}^{(k)} \tag{6.17}$$

We begin with $I_{n+1}^{(k)}(q)$. Given any tree T on the vertex set $[n+1]'$ rooted at 0, there is a unique edge e emerging from 0 that is on the path between the vertex 0 and the vertex 1. If we remove e from T, then we get two trees, T_0 and (T_1, r), where $e = \{0, r\}$. Suppose that the tree T_0 has $a_k + 1$ vertices. Then T_1 has $n + 1 - a_k$ vertices and

$$\mathrm{inv}_k(T) = \mathrm{inv}_k(T_0) + \mathrm{inv}_k(T_1, r)$$

since the colour of e is necessarily 0. Notice that T_0 is a rooted k-tree rooted at 0. Hence,

$$\begin{aligned} I_{n+1}^{(k)}(q) &= \sum_T q^{\mathrm{inv}_k(T)} \\ &= \sum_{(T_0,(T_1,r))} q^{\mathrm{inv}_k(T_0)+\mathrm{inv}_k(T_1,r)} \\ &= \sum_{a_k=0}^{n} \binom{n}{a_k} \sum_{T_0} q^{\mathrm{inv}_k(T_0)} \cdot \sum_{(T_1,r)} q^{\mathrm{inv}_k(T_1,r)} \\ &= \sum_{a_k=0}^{n} \binom{n}{a_k} I_{a_k}^{(k)}(q) \sum_{(T_1,r)} q^{\mathrm{inv}_k(T_1,r)} \end{aligned} \tag{6.18}$$

Now we compute $\sum_{(T_1,r)} q^{\mathrm{inv}_k(T_1,r)}$. Notice that as far as it pertains to tree inversion, the actual labels on the trees don't matter, only the relative ordering of the labels are important. Hence, we assume that all the trees in the sum over all (T_1, r) in (6.18) are on the vertex set $[n - a_k]'$. Let T_r be the trees with root r (see Figure 6.4). If we remove r from the tree T_1 we create a collection of rooted trees. Let $(H_{i,s_1}, r_1), (H_{i,s_2}, r_2), \ldots, (H_{i,s_i}, r_i)$, for $0 \leq i \leq k-1$, be the rooted trees that had edges coloured i connecting them to r. Suppose that the total number of vertices in all these trees is a_i and let S_i be the rooted tree created by attaching a new root 0 to all the roots of $(H_{i,s_1}, r_1), (H_{i,s_2}, r_2), \ldots, (H_{i,s_i}, r_i)$ (see Figure 6.4). Then

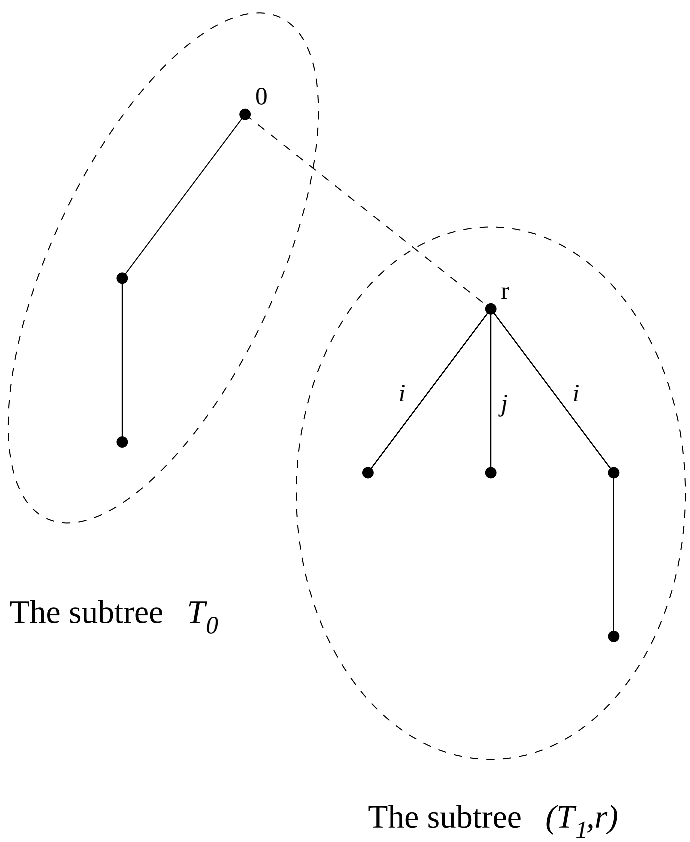

Figure 6.4: Decomposing a tree T into T_0 and (T_1, r).

clearly,

$$\mathrm{inv}_k(T_1,r) = r-1+a_1+2a_2+\cdots+ka_k+\mathrm{inv}_k(S_0)+\mathrm{inv}_k(S_2)+\cdots+\mathrm{inv}_k(S_{k-1})$$

implying that

$$\begin{aligned}
\sum_{(T_1,r)} q^{\mathrm{inv}_k(T_1,r)} &= \sum_{j=1}^{n+1-a_k} \sum_{T\in T_r} q^{\mathrm{inv}_k(T)} \\
&= \sum_{j=1}^{n+1-a_k} q^{r-1} \sum_{T\in T_r, |S_i|=a_i} q^{a_1+2a_2+\cdots ka_k+\mathrm{inv}_k(S_0)+\mathrm{inv}_k(S_2)+\cdots+\mathrm{inv}_k(S_{k-1})} \\
&= \sum_{j=1}^{n+1-a_k} q^{r-1} \sum_{\substack{a_0+a_1+\cdots a_{k-1}=n-a_k \\ |S_i|=a_i \text{ for } 0\le i\le k-1}} \binom{n-a_k}{a_0,a_1,\ldots,a_{k-1}} q^{a_1+2a_2+\cdots+ka_k} \times \\
&\qquad q^{\mathrm{inv}_k(S_0)+\mathrm{inv}_k(S_1)+\cdots+\mathrm{inv}_k(S_{k-1})} \\
&= \left(\sum_{j=0}^{n-a_k} q^j\right) \sum_{a_0+a_1+\cdots+a_{k-1}=n-a_k} \binom{n-a_k}{a_0,a_1,\ldots,a_{k-1}} \times \\
&\qquad q^{a_1+2a_2+\cdots+ka_k}\cdot I_{a_0}^{(k)}(q)\cdot I_{a_1}^{(k)}(q)\cdots I_{a_k}^{(k)}(q)
\end{aligned}$$

Substituting the last equation into (6.18) we get our result. We now prove (6.17).

Suppose that $G_0, G_1, \ldots, G_k$ is a sequence of connected k-graphs with $a_i+1 = |V(G_i)|$. Further, suppose that their vertex sets are disjoint except that each graph has a vertex, its root, labelled 0 and the union of all the vertex sets is $[n]'$. We want to "merge" the above graphs into one graph on $[n+1]'$. First, for $0\le i\le k-1$, in graph G_i colour the edges that emerge from 0 with the colour i. Second, for each i listed above, in G_i draw any number of edges from 0 to the other vertices of G_i having any colour less than $i-1$. Now, merge the graphs $G_0, G_1, \ldots, G_{k-1}$ into one connected k-graph by identifying all the vertices labelled 0 and call the new graph H. In H, relabel the vertex 0 with the label $n+1$. Now, make a new vertex 0 and attach any number of edges from 0 to the vertices H (no multiple edges) and call this new graph H'. All such edges will be given the colour 0 (this is why we did not allow multiple edges). Finally, merge the vertex 0 of G_k and the vertex 0 of H' to get a connected k-graph on $[n+1]'$.

The above procedure gives a bijection (it can clearly be reversed) between connected k-graphs on $[n+1]'$ and sequences $(G_0, G_1, \ldots G_k, E_0, E_1, \ldots, E_{k-1}, S)$ where the G's are given above, the E_i is a set of arbitrary edges whose endpoints are the root of G_i and some other vertex in G_i and S is a nonempty subset of the vertices in $G_0, G_1, \ldots, G_{k-1}$ to which we attached the new 0. Hence, $C^{(k)}_{n+1}(q)$ equals

$$\begin{aligned}\sum_G q^{e(G)-n} &= \sum_{a_0+a_1+\cdots+a_k=n} \sum_{\substack{(G_0,G_1,\ldots,G_k,E_0,E_1,\ldots,E_k,S)\\ |V(G_i)|=a_i}} q^{|E_0|}q^{|E_1|}\cdots q^{|E_{k-1}|}q^{|S|} \times \\ &\qquad q^{e(G_0)-a_0}\cdot q^{e(G_1)-a_1}\cdots q^{e(G_k)-a_k} \\ &= \sum_{a_0+a_1+\cdots+a_k=n} \binom{n}{a_0, a_1, \ldots a_k}(1+q)^{a_1}\cdots(1+q)^{(k-1)a_{k-1}} \times \\ &\qquad ((1+q)^{a_0+a_1+\cdots+a_{k-1}}-1)C^{(k)}_{a_0}(q)C^{(k)}_{a_1}(q)\cdots C^{(k)}_{a_k}(q)\end{aligned}$$

Dividing both sides by q completes the proof. □

From this it follows that the number of k-parking functions of length n is the number of rooted k-trees (with root 0) on $n+1$ vertices. If we define

$$R_k(x) = \sum_{i\geq 0} r_k \frac{x^n}{n!}$$

where r_k is the number of k-trees it is clear that $R_k(x)$ is equal to $\exp(y)$ where $y = x\exp(ky)$. Using the Lagrange Inversion formula (see Goulden and Jackson [7, Sec. 1.2] Stanley [24, Sec. 5.4]) gives $r_k = (kn+1)^{n-1}$. Compare this with Proposition 6.9.

Above, in Theorem 6.11 we showed that $P_n^{(k)}(q) = I_n^{(k)}(q)$ via generating function techniques and using connected k-graphs and the generating function $C_n^{(k)}(q)$. We now prove that $P_n^{(k)}(q) = I_n^{(k)}(q)$ by presenting a direct bijection. For clarity, we give an example after the proof and it is advised that one follows it along with the proof.

Theorem 6.12 *There exists a bijection θ_n between k-parking functions of length n and rooted k-trees with $n+1$ vertices that preserves the weights in the generating functions $P_n^{(k)}(q)$ and $I_n^{(k)}(q)$ i.e., the existence of θ_n implies $P_n^{(k)}(q) = I_n^{(k)}(q)$.*

Proof. We prove this by inductively defining a function θ_n. For the base case, $n = 0$, we map the empty k-parking function to the tree with the lone vertex 0.

Suppose that θ_n has been defined for all $n < m$. We now define θ_m. Let $p = (a_1, a_2, \ldots, a_m)$ be a k-parking function and let ω be the unique permutation such that 1) $a_{\omega(1)} < a_{\omega(2)} < \cdots < a_{\omega(m)}$ (i.e. $(a_{\omega(1)}, a_{\omega(2)}, \ldots, a_{\omega(m)})$ is a non-decreasing rearrangement of $(a_1, a_2, \ldots, a_m)$) and 2) if $a_{\omega(i)} = a_{\omega(j)}$ and $i < j$ then $\omega(i) < \omega(j)$. Suppose that s is the index such that $\omega(s) = m$ and let j be the largest index smaller than s that satisfies $a_{\omega(j)} = k(j-1)$. Define

$$A_1 = \{\ i \mid a_i < k(j-1)\}$$

and

$$A_2 = \{\ i \mid a_i > k(j-1)\}$$

The fact that $|A_1 \cup A_2| = m-1$ is guaranteed by Lemma 6.7. Define $p' = (a_i)_{i \in A_1}$. It is clear that p' is a k-parking function of length $j-1$ and, thus, $\theta_{j-1}(p')$ gives us a rooted k-tree T_1 (with labels in $A_1 \cup \{0\}$, by the induction hypothesis). We are further guaranteed by the induction hypothesis that

$$\mathrm{inv}_k(T_1) = k\binom{j-1}{2} - \sum_{i \in A_1} a_i$$

Dealing with the rest of p, we notice that $(a_{\omega(j+1)} - k(j-1)+1, a_{\omega(j+2)} - k(j-1)+1, \ldots, a_{\omega(s)} - k(j-1)+1, a_{\omega(s+1)} - k(j-1), \ldots, a_{\omega(m)} - k(j-1))$ is a $\hat{k}$-parking function written in non-decreasing order (we know that this is in non-decreasing order because $a_{\omega(s)} < a_{\omega(s+1)}$. Setting $r = s-j$, we wish to re-index the last sequence with the set A'_2 defined as $A_2 \backslash \{\omega(t)\}$, where $\omega(t)$ is the member of $A_2 \cup \omega(j)$ with rank $r+1$. To do this we consider the sequence of labels that currently label the sequence, namely $(\omega(j+1), \omega(j+2), \ldots, \omega(m))$. We create the new sequence $\beta = (\omega(j), \omega(j+1), \ldots, \omega(s-1), \omega(s+1), \ldots, \omega(m))$ (we removed the entry $\omega(s)$, shifted the entries $\omega(j+1), \omega(j+2), \ldots, \omega(s-1)$ to the right and made $\omega(j)$ the first entry). For all the entries in β, let $\omega(i) \oplus 1$ be the member of $A_2 \cup \omega(j)$ with next highest rank. Notice that we need not concern ourselves with what $\omega(s) \oplus 1$ equals since it does not occur in β. For all $i \geq j$ where rank of $\omega(i) > r+1$ we replace $\omega(i)$ in β with $\omega(i) \oplus 1$. Call this new vector β' which we suppose is $(\beta'_1, \beta'_2, \ldots, \beta'_{m-j})$. We now re-index the sequence $(a_{\omega(j+1)}+1, a_{\omega(j+2)}+1, \ldots, a_{\omega(s)}+1, a_{\omega(s+1)}, \ldots, a_{\omega(m)})$ with β'. Or, we can simply define a new sequence $(a''_{\beta'_1}, a''_{\beta'_2}, \ldots, a''_{\beta'_{m-j}})$ such that $a''_{\beta'_1} = a_{\omega(j+1)}+1, a''_{\beta'_2} = a_{\omega(j+2)}+1, \ldots, a''_{\beta'_{m-j}} = a_{\omega(m)}+1$ and define our $\hat{k}$-parking function to be $p'' = (a''_i)_{i \in A'_2}$ (notice that $(a_{\omega(j+1)} - k(j-1)+1, a_{\omega(j+2)} - k(j-1) + 1, \ldots, a_{\omega(s)}+1, a_{\omega(s+1)}, \ldots, a_{\omega(m)})$ is the non-decreasing rearrangement of p''). At this point we would also like to record the number $\omega(t)$ (as this is going to give us the root of the tree that we get from p'') and we emphasize this by writing $(p'', \omega(t))$. Further, we note that the above process is reversible, i.e. given a $\hat{k}$-parking function indexed on some set A and a number $\omega(t)$ not in A we can construct the sequence $(a_{\omega(j)}, a_{\omega(j+1)}, \ldots, a_{\omega(m)})$.

From Proposition 6.5 we have that $\hat{\theta}_{m-j}(p'')$ is a rooted $\hat{k}$-tree T_2 (rooted at 0 and with the rest of the labels coming from A'_2) such that

$$\mathrm{inv}_k(T_2) = k\binom{m-j+1}{2} - \sum_{i \in A'_2} a''_i$$

If we are to replace the label on the root (0) with the label $\omega(t)$ we obtain a new tree $(T_2', \omega(t))$ such that

$$\begin{aligned}
\mathrm{inv}_k(T_2', \omega(t)) &= \mathrm{inv}_k(T_2) + r \\
&= k\binom{m-j+1}{2} - \sum_{i \in A_2'} a_i'' + r \\
&= k\left(\sum_{i=1}^{m-j} i\right) - \sum_{i=j+1}^{s} (a_i - k(j-1) + 1) - \sum_{i=s+1}^{m} (a_i - k(j-1)) + r \\
&= k\left(\sum_{i=1}^{m-j} i\right) - \sum_{i \in A_2} (a_i - k(j-1)) \\
&= k\left(\sum_{i=j}^{m-1} i\right) - \sum_{i \in A_2} a_i \\
&= k\left(\sum_{i=j-1}^{m-1} i\right) - \sum_{i \in A_2 \cup \{\omega(j)\}} a_i
\end{aligned}$$

We now create the final tree T by attaching the vertex 0 of T_1 to the root $\omega(t)$ of T_2. We see that $\mathrm{inv}_k(T)$ equals

$$\begin{aligned}
\mathrm{inv}_k(T_1) + \mathrm{inv}_k(T_2', \omega(j)) &= k\binom{j-1}{2} - \sum_{i \in A_1} a_i + k\left(\sum_{i=j-1}^{m-1} i\right) - \sum_{i \in A_2 \cup \{\omega(j)\}} a_i \\
&= k\binom{m}{2} - \sum_{i=1}^{m} a_i
\end{aligned}$$

and since the above is reversible, this completes the proof. □

Instead of explicitly displaying the inverse we simply work out an example (using the notation in the proof) of both the forward map and the inverse.

Example 6.13 Let $k = 4$ and $p = (a_1, a_2, \ldots, a_n) = (0, 5, 12, 3, 19, 13, 16, 13)$. In this case, $\omega = 14236975$ and, hence, $(a_{\omega(1)}, a_{\omega(2)}, \ldots, a_{\omega(n)})$ is $(0, 3, 5, 12, 13, 13, 16, 19)$

and $s = 6$ (where s in the proof is defined as $\omega(s) = n$. Since the largest j less than s such that $a_{\omega(j)} = k(j-1)$ is $j = 4$ we have that p splits up into the two sequences $p' = (0, 5, 3)$ indexed by the set $A_1 = (1, 2, 4)$ and the sequence $(19, 13, 16, 13)$ indexed by the set $A_2 = (5, 6, 7, 8)$. $\omega(j) = \omega(4) = 3$ singles out the entry $a_3 = 12$. Applying induction to p' gives us the tree T_1 in Figure 6.5. Dealing with the rest of p, $r = s - j = 8 - 4 = 2$ and the member of $A_2 \cup \{\omega(j)\}$

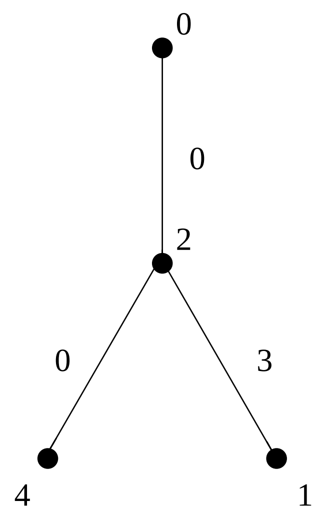

Figure 6.5: The tree T_1 in Example 6.13.

of rank $r + 1 = 3$ is 6, which we called $\omega(t)$ in the proof. The sequence of leftover labels $A_2 \cup \{\omega(j)\} = (\omega(j), \omega(j+1), \ldots, \omega(n)) = (3, 6, 8, 7, 5)$ and the first step of our re-indexing gives us $\beta = (3, 6, 7, 5)$ and finally we have $\beta' = (3, 7, 8, 5)$ with $\omega(t) = 6$. Hence, we obtain from β' the sequence $(a''_3, a''_7, a''_8, a''_5) = (13 - k(j-1) + 1, 13 - k(j-1) + 1, 16 - k(j-1), 19 - k(j-1)) = (2, 2, 4, 7)$ making a $\hat{k}$-parking function $p'' = (2, 7, 2, 4)$ indexed by $(3, 5, 7, 8)$. From Proposition 6.5 we get the tree T_2 from p'' given in Figure 6.6. Finally, the proof says to replace the root in

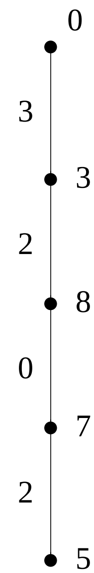

Figure 6.6: The tree T_2 in Example 6.13.

the tree T_2 with the label $\omega(t) = 6$ and attach the vertex 6 to the vertex 0 of T_1. The final tree T is in Figure 6.7. Indeed,

$$k\binom{n}{2} - \sum_{i=1}^{8} a_i = 112 - 81 = 31$$

and

$$\mathrm{inv}_K(T) = 31$$

To reverse the process, suppose that we are given the tree T in Figure 6.7. We find the largest label, 8 in our case, in the tree and we remove the edge $\{0, 6\}$ from the tree ($\{0, 6\}$ being the edge emerging from 0 on the unique path from 0 to 8). We obtain two trees T_1 and $(T_2', 6)$. Define A_1 and A_2 to be the vertex sets of T_1 and

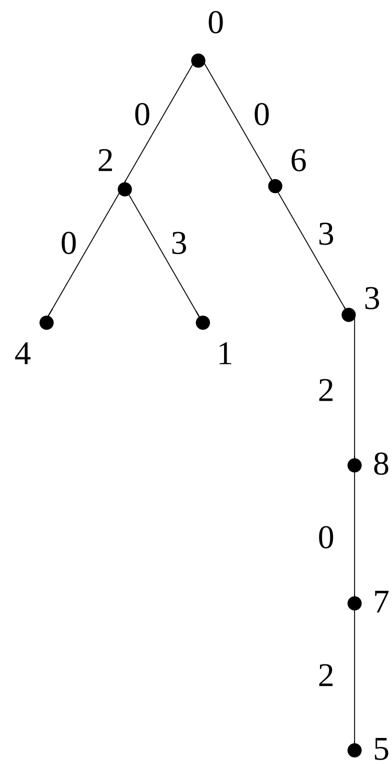

Figure 6.7: The image of p under θ_8 in Example 6.13.

$(T_2', 6)$, respectively. By the induction hypothesis, there exists a k-parking function p' such that $\theta_{j-1}(p') = T_1$. We index p' with the elements of A_1 and in this case we get

$$p' = (a_1, a_2, a_4) = (0, 5, 3) \tag{6.19}$$

For $(T_2', 6)$ we replace the root label 6 with the label 0, obtaining a tree T_2. From Proposition 6.5 we get a $\hat{k}$-parking function indexed by $A_2 \backslash \{6\}$, namely $p'' = (a_3'', a_5'', a_7'', a_8'') = (2, 7, 2, 4)$. We must now subtract 1 from the 2 smallest members of p'' (the number 2 is obtained from the fact that 2 is the rank of the root 6 in the set A_2 minus 1). This gives us the sequence $\bar{p}'' = (a''_3 - 1, a''_5, a''_7 - 1, a''_8) = (1, 7, 1, 4)$. Next, we wish to re-index p''. We do this by reversing the relabelling done in the above proof. We let $\beta' = (3, 7, 8, 5)$ (the order of the labels when we write p'' in non-decreasing order). The rank of 6 in the set A_2 is 3, implying that we perform the operation $\ominus 1$ on every label in $(3, 7, 8, 5)$ of rank 3 or higher (where $\ominus 1$ is the obvious inverse of the operation $\oplus 1$ in the proof) and obtain the sequence $\beta = (3, 6, 7, 5)$. We now insert the highest label (8) into the 2^{nd} spot of β, move the label in spot 2 one to the left and remove the label 3 from the sequence. Doing this gives us the sequence $(6, 8, 7, 5)$. Now we add $k(j-1) = 12$ to each member of $\bar{p}''$ above (where j is the cardinality of A_1). Define the new sequence

$$(a_6, a_8, a_7, a_5) = (13, 13, 16, 19) \tag{6.20}$$

Set $a_3 = k(j-1) = 12$ (where 3 is the label we removed from β). Finally, we combine (6.19) and (6.20) with $a_3 = 12$ and we get $p = (a_1, a_2, \ldots, a_n) = (0, 5, 12, 3, 19, 13, 16, 13)$, the k-parking function we started with. □

Notes and References

§6.1 The bijection involving noncrossing partitions is due to R. Stanley and can be found in [23, Thm. 3.1]. There Stanley gives the proof for $k = 1$, and the extension to arbitrary k was essentially routine. Concerning rooted $\hat{k}$-trees, we note that they are not mentioned anywhere in the current literature and they were basically introduced because of their use in Theorem 6.12. As we mentioned earlier,

$\hat{k}$-parking functions are a routine extension of parking functions and aren't nearly as interesting or complex as k-parking functions.

§6.2 Most of the material in this section can be found in Stanley [25] and Yan [27]. The proof of Theorem 6.8 is in [27] (although there the generating function is for "complements of parking functions") from which it was just a matter of working with generating functions to obtain the number of k-parking functions, given in Proposition 6.9. Theorem 6.11 can be found in both [25][27]. It is this author's opinion that the treatment of Theorem 6.11 in [27] is slightly more accessible than in [25]. Theorem 6.12 was given by the author of this thesis, with the help of I.P. Goulden.

Although little was made of the generating function $C_n^{(k)}(q)$, it was useful in the preceding section because it satisfied the two recursions given in Theorem 6.11. What is more remarkable is that in [27], Yan gives yet another interesting recursion satisfies by $C_n^{(k)}(q)$ (it is given without proof).

Bibliography

[1] C. Athanasiadis. A simple bijection for the regions of the Shi arrangement of hyperplanes. *Discrete Mathematics*, 204:27–39, 1999.

[2] J. S. Beissinger and U. N. Peled. A note on major sequences and external activity. *Electronic Journal of Combinatorics*, 4(2), 1997.

[3] J.D. Dixon and B. Mortimer. *Permutation Groups.* Springer, New York, 1996.

[4] P.H. Edelman. Chain enumeration and noncrossing partitions. *Discrete Mathematics*, 31:171–180, 1980.

[5] D. Foata and J. Riordan. Mappings of acyclic and parking functions. *Aequationes Mathématique*, 10:10–22, 1974.

[6] I. Gessel and D.-L. Wang. Depth-first search as a combinatorial correspondence. *Journal of Combinatorial Theory, (A)*, 26:308–313, 1979.

[7] I. P. Goulden and D. M. Jackson. *Combinatorial Enumeration.* John Wiley and Sons, 1983.

[8] P. Headley. *Reduced expressions in infinite Coxeter groups.* PhD thesis, University of Michigan, Ann Arbor, 1994.

[9] A. G. Konheim and B. Weiss. An occupancy discipline and applications. *SIAM Journal of Applied Mathematics*, 14:1266–1274, 1966.

[10] G. Kreweras. Une famille de polynômes ayant plusieurs propriétés énumeratives. *Periodica Mathematica Hungarica*, 11:309–320, 1980.

[11] W. Ledermann. *Introduction to Group Characters.* Cambridge University Press, Cambridge, 1977.

[12] I. G. Macdonald. *Symmetric Functions and Hall Polynomials.* Oxford University Press, Oxford, second edition edition, 1995.

[13] C. L. Mallows and J. Riordan. The inversion enumerator for labeled trees. *Bulletin of the American Mathematics Society*, 74:92–94, 1968.

[14] A. Nica and R. Speicher. A "Fourier transform" for multiplicative functions on noncrossing partitions. *Journal of Algebraic Combinatorics*, 6:141–160, 1997.

[15] B. E. Sagan. *The Symmetric Group: Representations, Combinatorial Algorithms and Symmetric Functions.* Wadsworth and Brook/Cole, Pacific Grove, California, 1991.

[16] J. Y. Shi. *The Kazhdan-Lutsztig cells in certain affine Weyl groups.* Number 1179 in Lecture Notes in Mathematics. Springer, Berlin/Heidelberg/New York, 1986.

[17] J. Y. Shi. Sign types corresponding to an affine Weyl group. *Journal of the London Mathematics Society*, 35:56–74, 1987.

[18] R. Simion and D. Ullman. On the structure of the lattice of noncrossing partitions. *Discrete Mathematics*, 98:193–206, 1991.

[19] R. Speicher. Multiplicative functions on the lattice of noncrossing partitions and free convolution. *Mathematische Annalen*, 298:611–628, 1994.

[20] R. P. Stanley. *Enumerative Combinatorics*, volume 1. Cambridge University Press, Cambridge, 1996.

[21] R. P. Stanley. Flag-symmetric and locally rank-symmetric partially ordered sets. *Electronic Journal of Combinatorics*, 3, 1996.

[22] R. P. Stanley. Hyperplane arrangements, interval orders, and trees. *Proceedings in the National Academy of Sciences*, 93, 1996.

[23] R. P. Stanley. Parking functions and noncrossing partitions. *Electronic Journal of Combinatorics*, 4, 1997.

[24] R. P. Stanley. *Enumerative Combinatorics*, volume 2. Cambridge University Press, 1998.

[25] R. P. Stanley. Hyperplane arrangement, parking functions and tree inversions. In *Mathematical Essays in Honor of Gian-Carlo Rota*, Progress in Mathematics, pages 359–375. Birkhäuser Boston, Boston, MA, 1998.

[26] H. Wilf. *Generatingfunctionology*. Academic Press, Boston, 2^{nd} edition, 1994.

[27] C. H. Yan. Generalized tree inversions and k-parking functions. *Journal of Combinatorial Theory Ser. A*, 79(No. 2):268–280, 1997.