

PARKING FUNCTIONS AND GENERALIZED CATALAN NUMBERS

A Dissertation

by

PAUL R. F. SCHUMACHER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2009

Major Subject: Mathematics

PARKING FUNCTIONS AND GENERALIZED CATALAN NUMBERS

A Dissertation

by

PAUL R. F. SCHUMACHER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Catherine Yan
Committee Members,	Marcelo Aguiar
	Nancy Amato
	Sue Geller
Head of Department,	Albert Boggess

August 2009

Major Subject: Mathematics

ABSTRACT

Parking Functions and Generalized Catalan Numbers. (August 2009)

Paul R. F. Schumacher, B.A.;B.A.;B.A., University of Chicago

Chair of Advisory Committee: Dr. Catherine Yan

Since their introduction by Konheim and Weiss, parking functions have evolved into objects of surprising combinatorial complexity for their simple definitions. First, we introduce these structures, give a brief history of their development and give a few basic theorems about their structure. Then we examine the internal structures of parking functions, focusing on the distribution of descents and inversions in parking functions. We develop a generalization to the Catalan numbers in order to count subsets of the parking functions. Later, we introduce a generalization to parking functions in the form of k -blocked parking functions, and examine their internal structure. Finally, we expand on the extension to the Catalan numbers, exhibiting examples to explore its internal structure. These results continue the exploration of the deep structures of parking functions and their relationship to other combinatorial objects.

To my parents, who never stopped believing in me,
and to all of the great teachers who inspired me along the way.

ACKNOWLEDGMENTS

I am grateful to Texas A&M University for giving me the support necessary to complete this project. Specifically, I would like to acknowledge Dr. Catherine Yan for her guidance along the way, Dr. Sue Geller for proofreading help, and my committee for their support and patience.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION TO PARKING FUNCTIONS	1
	A. History of Parking Functions	1
	B. Structure and Relationships of Parking Functions	2
II	DESCENTS AND INVERSIONS IN PARKING FUNCTIONS	7
	A. Descents in Parking Functions	7
	1. Counting Descents	7
	2. Descents and Unicyclic Graphs	11
	B. Weighted Ballots	15
	1. The Ballot Problem	15
	2. One-sided Generalization	16
	3. Plane Trees of a Given Type	18
	4. Weighted Schröder Ballot Arrangements	20
	C. Inversions in Parking Functions	21
	1. Parking Functions and Permutations	21
	2. Counting Inversions	22
III	<i>K</i> -BLOCKED PARKING FUNCTIONS	25
	A. Introduction	25
	B. Blocked Parking Functions and Labeled Schröder Paths	28
	C. <i>p</i> -Parking Functions	29
	D. <i>p</i> -Parking Functions With <i>k</i> Blocked	30
	E. Linear Probes in $PB_{(n,k)}$	31
IV	LEFT WEIGHTED CATALAN STRUCTURES	40
	A. Trees	41
	B. Lattice Sequences	45
	C. Partitions	49
	D. Permutations	56
	E. Other Equivalent Representations	61
	F. Two-Sided Generalization	62
V	SUMMARY	64

Page

REFERENCES 65

VITA 68

LIST OF FIGURES

FIGURE		Page
1	Parking function distribution for $n = 6$	8
2	$T3_4$	12
3	Ballot problem for $n = 3$	15
4	3-Catalan ballot problem	16
5	Tree for $22 - - - 1 - -$	19
6	Weighted ballots (Catalan case (\mathbf{r})) $\mathbb{B} = \{1^1, 2^2\}$	41
7	Weighted Calatan examples for (\mathbf{c})	42
8	Weighted Calatan examples for (\mathbf{d})	42
9	Weighted Calatan examples for (\mathbf{e})	43
10	Weighted Calatan examples for (\mathbf{f})	44
11	Weighted Calatan examples for (\mathbf{g})	44
12	Weighted Calatan examples for (\mathbf{h})	46
13	Weighted Calatan examples for (\mathbf{i})	46
14	Weighted Calatan examples for (\mathbf{j})	47
15	Weighted Calatan examples for (\mathbf{k})	47
16	Weighted Calatan examples for (\mathbf{l})	48
17	Weighted Calatan examples for (\mathbf{r}^*)	50
18	Weighted Calatan examples for (\mathbf{o})	50
19	Weighted Calatan examples for (\mathbf{n})	51

FIGURE		Page
20	Weighted Calatan examples for (p)	52
21	Weighted Calatan examples for (a)	53
22	Weighted Calatan examples for (pp)	54
23	Weighted Calatan examples for (rr)	55
24	Weighted Calatan examples for (uu)	55
25	Weighted Calatan examples for (cc)	56
26	Weighted Calatan examples for (dd)	57
27	Weighted Calatan examples for (gg)	59
28	Weighted Calatan examples for (ii)	59
29	Weighted Calatan examples for (jj)	60
30	Weighted Calatan examples for (aaa)	61
31	Weighted Calatan examples for (b)	61
32	Weighted Calatan examples for (s)	62
33	Weighted Calatan examples for (ww)	62

CHAPTER I

INTRODUCTION TO PARKING FUNCTIONS

A. History of Parking Functions

Parking functions were introduced in 1966 by Konheim and Weiss[13]. The original concept was that of a linear parking lot with n available spaces, and n cars with a stated parking preference. Each car would, in order, attempt to park in its preferred spot. If the car found its preferred spot occupied, it would move to the next available slot. A parking function is a sequence of parking preferences that would allow all n cars to park using this method.

Definition A.1. *Formally, a parking function of length n is a map from $[n]$ to $[n]$ (written $(a_1 a_2 \dots a_n)$) such that for all $i \leq n$, the number of $a_j : a_j \leq i$ is greater than or equal to i .*

(Alternatively, if b_i are the a_i sorted into a non-decreasing order, then $b_i \leq i$.)

We will refer to the set of parking functions of length n as PF_n (see B.1 for the size of PF_n). Foata and Riordan found that the parking functions of size n are in bijection with labeled trees with $n+1$ nodes and Stanley found a relationship between parking functions and non-crossing partitions[23]. Further relationships have been found to other structures, such as hyperplane arrangements[24], priority queues[5], and Goncarov polynomials[15].

Knuth explored parking functions as a hashing problem with a linear collision resolution[11]. Kreweras examined the number of linear probes in parking functions (using slightly different terminology), the number of spaces each car has to bypass

This dissertation follows the style of Mathematics of Computation.

in order to park. Kreweras presented a polynomial family which gives the number of linear probes required for the functions in PF_n [14]. Beissinger and Peled linked Kreweras' polynomial family to inversions and external activity in labeled trees[1]. Knuth further explored linear probing as well[12].

The base concept of parking functions has also been extended to more generalized objects. Stanley proposed “k-parking functions”, which consist of a map from $[n]$ to $[kn]$ such that for all $i \leq n$, the number of $a_j \leq (i-1)k+1$ is greater than or equal to i , and this concept was further expanded and explored by Yan[28]. (Alternatively, if b_i are the a_i arranged in a non-decreasing order, then $b_i \leq (i-1)k+1$.) Gilbey and Kalikow added “types” with a preferred subset of spaces to create valet functions[5]. Biane generalized to parking functions of “Type B” where the function only required $a_i \leq n$, and showed that this generalized the relationship between parking functions and non-crossing partitions given earlier by Stanley[2]. Loehr, Haglund, and Haiman have explored the connections between parking functions and q, t -Catalan structures, as well as a variety of algebraic structures (see [6][7][19] for examples). Yan has done further exploration of various aspects of parking functions and their relation to other structures (Yan and Kostic[31], Kung and Yan[16, 17], and Yan[29, 30, 18]).

B. Structure and Relationships of Parking Functions

Remark. The definition of “degree” varies when applied to graphs and trees. For reasons of clarity, we will define the degree of a node throughout to be the number of children a node has in a rooted tree, and the valence of a node to be the number of nodes connected to it, whether it is in a tree or a graph.

Theorem B.1. *There are $(n+1)^{n-1}$ parking functions in PF_n .*

Proof. 1: Given a sequence of $(a_1 a_2 \dots a_n)$ such that $a_i \in [n+1]$, we define $a'_1 = a_1$

and a'_i to be the first element of the sequence $(a_i, a_i + 1 \pmod{n+1}, a_i + 2 \pmod{n+1}, \dots)$ that isn't in $\{a'_1, a'_2, \dots, a'_{i-1}\}$. Let C_n be the set of maps $\{f : [n+1] \rightarrow [n]\}$. There are $(n+1)^n$ such maps. Now we define an map $\phi : C_n \rightarrow PF_n$ such that if $f = (f_1, f_2, \dots, f_n) \in C_n$, then $\phi(f) = (f_1 - f_0 \pmod{n}, f_2 - f_0 \pmod{n}, \dots, f_n - f_0 \pmod{n})$, where f_0 is the sole element of $[n+1]$ not in $\{a'_1, \dots, a'_n\}$. For each member of PF_n , there will be $n+1$ precursors in C_n , dependent solely on the value of f_0 , which can range from 0 to n . Thus, each equivalence class has $n+1$ members, and so $\#PF_n = \frac{(n+1)^n}{n+1} = (n+1)^{n-1}$. (Konheim and Weiss[13].) \square

Proof. 2: Given a parking function $f = (a_1 a_2 \dots a_n) \in PF_n$, define the Prüfer code of f to be $(a_2 - a_1 \pmod{n+1}, \dots, a_n - a_{n-1} \pmod{n+1})$. Given the transitions of the parking function, the first element can be uniquely determined which will make it a valid parking function; We choose a starting point on a circle with $n+1$ slots and “Park” the function around it. Then we label the slots so that the empty slot is marked 0 and the rest of the slots are numbered consecutively. This gives us a unique first element of our parking function. Thus, each parking function is uniquely determined by the Prüfer code it maps to. Since the Prüfer codes are the maps from $[n-1] \rightarrow [n+1]$, we know that there are $(n+1)^{n-1}$ of them. (Foata and Riordan[4].) \square

The Prüfer codes given by Foata and Riordan are important when exploring many aspects of parking functions, since they reduce the parking function to the distances between successive elements. They also lead to a deeper understanding of the internal structure of the parking functions, and their relationship to other combinatorial objects, such as trees.

Corollary B.2. $PF_n \cong LT_{n+1}$, the set of labeled trees on $n+1$ nodes.

Proof. Each parking function can be transformed into a corresponding Prüfer code, as shown above. Each Prüfer code corresponds to exactly one labeled tree, as follows: Given a tree labeled 0 to n , remove the node of valence one with the highest label, and note which node it was removed from. Repeat this process until only two nodes remain, forming a sequence of $n-1$ removals. The remaining two nodes will necessarily be the 0 node and the last referenced node in the sequence other than 0. (If all of the previous nodes were removed from 0, the non-zero node must necessarily be the smallest non-zero node in the tree, *i.e.*, 1.) This gives us the Prüfer code for the tree. For any sequence of $[n-1]$ elements of $[n+1]$, we get exactly one tree. Since we already have a bijection from the parking functions to the Prüfer codes, this gives us a bijection to the labeled trees as well. (Foata and Riordan[4]) \square

Remark. The algorithm for generating a Prüfer code for a tree naturally removes nodes of valence one from the tree until only the node labeled 0 remains. Because of this, by convention, we refer to the labeled trees as if they were rooted at the 0 node. This allows us to refer to a node's parent, which is either the 0 node, or the node on a path between a given node and the 0 node.

There are many equivalent definitions for Dyck paths. We will use the following:

Definition B.3. *Given n , a Dyck path of length $2n$ is a set of n up steps $(0, 1)$ and n over steps $(1, 0)$, such that for any over step, the number of up steps preceding it is more than the number of over steps preceding it. (Equivalently, the path never falls below the diagonal $x = y$.)*

The number of unlabeled rooted trees with $n+1$ nodes and the number of Dyck paths of length $2n$ are both well known examples of the Catalan numbers[25]. Since we have related labeled trees to parking functions, and “rooted” them at the node

labeled zero, we naturally assume that there is equivalently, a labeled version of Dyck paths that are also equivalent to parking functions. We define the labeled Dyck paths of length $2n$ (LD_n) to be these paths with the up steps labeled with the elements of $[n]$ such that any up step immediately preceded by another up step has a higher label than the preceding up step.

Theorem B.4. $PF_n \cong LD_n$.

Proof. Let b_i be the number of i in the parking function f . We create a Dyck path with b_i up steps in column i . Then we label the i th column with the locations of i in the parking function, in ascending order. Since $c_i = \#\{a_j \leq i\} \geq i$, there are $c_i \geq i$ up steps before the i th over step.

Given a labeled Dyck path, let d_i be the number of up steps before the i th over step. By the definition of the labeled Dyck paths, $d_i \geq i$. If we use the inverse of the process above, this means that d_i elements of the parking function will be less than or equal to i for all i , and $d_i \geq i$, meaning we have a valid parking function. Thus, we have a map from $PF_n \rightarrow LD_n$ and exhibited an inverse. (N.A. Loehr[19]) \square

Corollary B.5. $LT_{n+1} \cong LD_n$

Proof. This is a combination of B.4 and B.2. \square

Let IP_n be the set of non-decreasing parking functions, *i.e.*, those parking functions $(a_1 a_2 \dots a_n)$ where $a_i \leq a_{i+1}$ for all $1 \leq i \leq n - 1$. Let D_n be the set of Dyck paths of length $2n$, and T_n be the set of rooted trees on n nodes.

Corollary B.6. $IP_n \cong D_n$.

Proof. Given any Dyck path, if we apply the trivial labeling where the labels occur in increasing order as we follow the path, we get a parking function where all of the elements equal to i come after those elements less than i . \square

Corollary B.7. $IP_n \cong T_{n+1}$.

Remark. This can be seen directly from the previous corollary, but the map below gives insight into the nature of the relationship that will be useful later.

Proof. We will create a map from the rooted trees to the non-decreasing parking functions, as follows: Given a rooted tree, create its degree sequence[25], as follows: let b_0 be the degree of the root, b_1 be the degree of the root's leftmost child, etc, so that b_i is the degree of the i th node of the tree, in pre-order traversal. Then b_i will give the number of copies of $i + 1$ that are in the parking function. Since the parking function is to be non-decreasing, there is only one distinguishable ordering of these elements.

The inverse map is clear: Given a non-decreasing parking function, the number of copies of $i + 1$ gives the degree of node i in pre-order traversal. \square

These relationships will be used later, along with our generalization of the Catalan numbers to help us count the number of parking functions with specific restrictions.

CHAPTER II

DESCENTS AND INVERSIONS IN PARKING FUNCTIONS

A. Descents in Parking Functions

1. Counting Descents

Let PF_n be the set of parking functions of length n .

Definition A.1. *Given a parking function $(a_1 a_2 \dots a_n)$, we look at a_i and a_{i+1} as a step in the parking function. We call this step a tie if $a_i = a_{i+1}$, a descent if $a_i > a_{i+1}$ and an ascent if $a_i < a_{i+1}$.*

There are $n - 1$ steps in each parking function. For example, in the parking function (1422), the step 14 is an ascent, the step 42 is a descent, and the step 22 is a tie.

Let $PF_{(n,i)} \subset PF_n$ be the set of parking functions of length n with i ties.

We can count the number of parking functions with i ties and j descents and arrange the results as a table such that i decreases as we read down the table and j decreases as we go from left-to-right. If we do this, we get a triangular set of numbers (see figure 1) with several interesting properties. The right side of the triangle shows the numbers of non-decreasing parking functions with i ties, from n ties at the top to 0 ties at the bottom. The left side, by symmetry shows non-increasing parking functions in the same way.

Corollary A.2. *If $IP_{(n,i)}$ is the set of non-decreasing parking functions with i ties of length n , and $D_{(n,j)}$ is the set of Dyck paths of length $2n$ with j peaks, then $IP_{(n,i)} \cong D_{(n,n-i)}$.*

Proof. Given any non-decreasing parking function with i ties, we note that there must

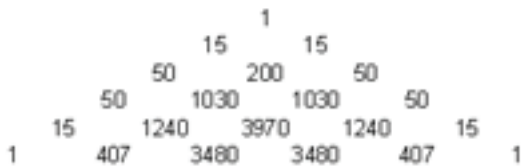


Fig. 1. Parking function distribution for $n = 6$

be exactly $n - i$ different elements in the parking function. When we use the map from I.B.4 to a Dyck path, we will get a Dyck path with exactly $n - i$ peaks. \square

Corollary A.3. *The number of parking functions of length n with no descents and i ties is $\frac{1}{i+1} \binom{n}{i} \binom{n-1}{i}$.*

Proof. Sloane's A001263[22], the triangle of Narayana numbers, gives us $T(n, j) = \frac{1}{j} \binom{n}{j-1} \binom{n-1}{j-1}$ as the number of Dyck paths with exactly j peaks. From above, we know that this is also the number of non-decreasing parking functions of length n with $n - j$ ties. Letting $i = n - j$, we get $\frac{1}{n-i} \binom{n-1}{n-i-1} \binom{n}{n-i-1} = \frac{1}{i+1} \binom{n-1}{i} \binom{n}{i}$. \square

Theorem A.4. *There are $\binom{n-1}{i} n^{n-1-i}$ parking functions in $PF_{(n,i)}$.*

Proof. Using the bijection in B.1, we note that each tie in a parking function becomes a 0 in the corresponding Prüfer code, and that any sequence of $[b_1 b_2 \dots b_{n-1}]$ in $\{0 \dots n\}^{n-1}$ is a valid Prüfer code. Therefore, if we want a parking function with i ties, we fix i zeroes in the code, and the other elements can be arbitrary non-zero elements of $[n]$. This gives us a total of $\binom{n-1}{i} n^{n-1-i}$ codes with exactly i zeroes, which, in turn, gives us the required count of parking functions in $PF_{(n,i)}$. \square

Corollary A.5. *There are $\binom{n-1}{i} n^{n-1-i}$ labeled trees with $n + 1$ nodes rooted at 0 such that the node labeled 0 has degree $i + 1$.*

Proof. In the Prüfer code for any labeled tree, a 0 in the code designates a node being removed that was connected to the node labeled 0. When the tree is down to two nodes, one of them is the zero node, and the other is removed from it. This last node removal is understood and thus not listed in the Prüfer code. This means that there were a total of $i + 1$ nodes attached to the zero node, where i is the number of zeroes in the Prüfer code for the tree. The number of such codes is counted in the proof of A.4. \square

Corollary A.6. *If we fix a label $a \in \{1, \dots, n\}$, there are $\binom{n-1}{i} n^{n-1-i}$ labeled trees with $n + 1$ nodes rooted at 0 such that the node labeled a has valence $i + 1$.*

Proof. We can create an automorphism on the set of labeled trees that switches the labels 0 and a . Therefore, the number of trees with node 0 having degree $i + 1$ is the same as the number of trees with label a having valence $i + 1$. \square

Corollary A.7. *The generating function of parking functions with j non-tie steps in PF_n is*

$$\sum_{i=0}^{n-1} \binom{n-1}{j} n^j x^j = (1 + nx)^{n-1}$$

.

Proof. This is a summation of the result from Theorem A.4 with $j = n - 1 - i$. \square

Lemma A.8. *The distribution of ascents and descents in PF_n and $PF_{(n,i)}$ are symmetrical: That is, they have the same number of ascents as descents.*

Proof. If we flip a parking function and look at $(a_n a_{n-1} \dots a_1)$ we see that we still have a parking function. In other words, this reordering is an automorphism of PF_n . However, under this automorphism, all ascents become descents, all descents

become ascents, and all ties remain ties. This symmetry tells us that the number of descents in PF_n must equal the number of ascents. Since the ties are unchanged, this automorphism also preserves the subsets $PF_{(n,i)}$ of PF_n , meaning that the number of descents in $PF_{(n,i)}$ must equal the number of ascents. \square

Corollary A.9. *There are $\frac{n-1-i}{2} \binom{n-1}{i} n^{n-1-i}$ descents in $PF_{(n,i)}$.*

Proof. Since there are $n-1$ steps for each parking function in $PF_{(n,i)}$, and i of each of these are ties, this leaves $n-1-i$ non-ties for each parking function in $PF_{(n,i)}$, and by the symmetry noted above, half of these are descents. \square

Theorem A.10. *If a_i is the number of descents in parking functions with i ties, then its generating function is $\sum_{i=0}^{n-1} a_i y^i = \binom{n}{2} (n+y)^{n-2}$.*

Proof. From A.9, we know that there are $\frac{n-1-i}{2} \binom{n-1}{i} n^{n-1-i}$ descents in $PF_{(n,i)}$, so we sum this number over i .

$$\sum_{i=0}^{n-1} \frac{n-1-i}{2} \binom{n-1}{i} n^{n-1-i} y^i$$

Setting $j = n-1-i$ gives us

$$\sum_{j=0}^{n-1} \frac{j}{2} \binom{n-1}{j} n^j y^{n-1-j} = \sum_{j=0}^{n-1} \frac{n}{2} \binom{n-1}{j} j n^{j-1} y^{n-1-j}$$

Temporarily replacing n^{j-1} with x^{j-1} yields

$$\left[\sum_{j=0}^{n-1} \frac{n}{2} \binom{n-1}{j} j x^{j-1} y^{n-1-j} \right]_{x=n}$$

$$\begin{aligned}
&= \frac{n}{2} \left[\frac{\delta}{\delta x} \left(\sum_{j=0}^{n-1} \binom{n-1}{j} x^j y^{n-1-j} \right) \right]_{x=n} \\
&= \frac{n}{2} \left[\frac{\delta}{\delta x} (x+y)^{n-1} \right]_{x=n} \\
&= \frac{n(n-1)}{2} (n+y)^{n-2} \\
&= \binom{n}{2} (n+y)^{n-2}
\end{aligned}$$

□

Corollary A.11. *There are $\binom{n}{2} (n+1)^{n-2}$ descents in PF_n .*

Proof. Plugging in $y = 1$ in A.10 gives us the desired result. □

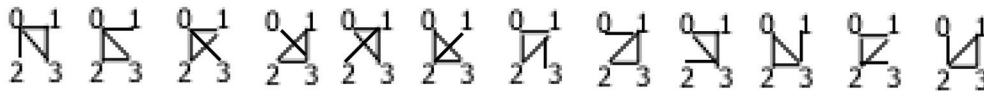
Corollary A.12. *The density of descents among steps is $\frac{n}{2(n+1)}$, or an average of $\binom{n}{2} \frac{1}{n+1}$ descents per parking function in PF_n .*

Proof. There are $(n-1)(n+1)^{n-1}$ steps in PF_n . $\binom{n}{2} (n+1)^{n-2}$ of these are descents. Division of the latter by the former gives us a density of $\frac{n}{2(n+1)}$. Multiplying by $n-1$ steps per parking function gives us the average number of descents per parking function. □

Remark. By symmetry, all of these results hold for ascents.

2. Descents and Unicyclic Graphs

Sloane's A053507[22] shows us another sequence with the same formula as the number of descents in PF_n . Let $T\mathfrak{3}_n$ be the set of unicyclic connected graphs on n nodes with

Fig. 2. $T3_4$

cycle length of three. (A **unicyclic graph** is one with only one cycle. Thus, $T3_n$ is the set of connected graphs with a single cycle of length three and no other cycles. See figure 2 for an example.) We came across this structure while searching for a proof of the number of descents in PF_n , and for completeness wish to exhibit a bijection between these structures. Since we have not yet disentangled the ascents and descents in PF_n , we do the next best thing and create a map from the union of the ascents and descents to two copies of $T3_n$.

Let \hat{A}_n be the set of ascents in PF_n and \hat{D}_n be the set of descents in PF_n . We choose to represent the elements of \hat{A}_n and \hat{D}_n as pairs (f, s) where f is a parking function in PF_n and s is the step in which the ascent or descent in question occurs. Using the map defined in corollary I.B.2, we can also represent this as $([b_1 b_2 \dots b_{n-1}], c)$ where the b_i give the Prüfer code for f . Define a new map, $\theta_n : \hat{A}_n \cup \hat{D}_n \rightarrow [n] \times [n-1] \times [n+1]^{n-2}$ such that $\theta_n(f, s) = (b_s, s, (b_1, \dots, b_{s-1}, b_{s+1}, \dots, b_{n-1}))$.

Lemma A.13. θ_n is a bijection

Proof. θ_n is one-to-one: Let (f, s) and (g, t) be distinct elements of $\hat{A}_n \cup \hat{D}_n$. Then either $f \neq g$, in which case, one of the b_i must be different between them (since ϕ is a bijection on PF_n) or $s \neq t$.

θ_n is onto: Given an arbitrary member of $[n+1]^{n-1}$, we know that ϕ^{-1} will give us a unique parking function. If we fix an s and restrict ourselves to elements of $[n+1]^{n-1}$ that have a non-zero element in the s th position, we have restricted ourselves to the subspace $[n+1]^{n-2} \times [n]$, but we see that we can still apply ϕ^{-1} to this element to get a parking function, and the restriction to a non-zero element in position s now gives us an ascent or descent in that step. Thus, $\theta_n^{-1}(b_s, s, (b_1, b_2, \dots, b_{s-1}, b_{s+1}, \dots, b_{n-1})) = (\phi^{-1}(b_1, \dots, b_{n-1}), s)|_{b_s \neq 0}$. \square

We can create Prüfer codes for the unicyclic graphs just as we did for the labeled trees: Given a unicyclic graph of cycle length three with labels from 0 to n , remove the node with largest label and degree one, recording it as b_1 . Repeat this process, removing the largest node of degree one successively to get $b_2 \dots b_{n-2}$, until we are left with the three-cycle. Record the nodes in the cycle as a set (noting that one of these members is also b_{n-2}).

Definition A.14. For an ordered set N , let the rank of $n \in N$ be one more than the number of $i \in N$ such that $i < n$.

We will define a map $\gamma_n : T_{n+1} \times \{0, 1\} \rightarrow [n-1] \times [n] \times [n+1]^{n-2}$. Given $(h, x) \in T_{n+1} \times \{0, 1\}$ let $\{a', b', c_{n-2}\}, (c_1, \dots, c_{n-2})$ be our Prüfer code for h , such that if $x = 0$, $a' < b'$ otherwise $a' > b'$. Then let b be the rank of b' in $[n+1] - \{c_{n-2}\}$ and a be the rank of a' in $[n+1] - \{c_{n-2}, b'\}$. This gives us elements of $[n]$ and $[n-1]$ respectively. The remainder of the Prüfer code is an element of $[n+1]^{n-2}$.

Lemma A.15. γ_n is a bijection.

Proof. γ_n is 1-1: Given any two elements of $T_{n+1} \times \{0, 1\}$, either the graph is different, which results in a different Prüfer code, or the element $\{0, 1\}$ is different, which results in different values of a and b .

γ_n is onto: Given a and b , we can recover the elements a' , b' and the element of $\{0, 1\}$ by comparing a' and b' . The remainder of the Prüfer code for our tree is unchanged by γ_n , and the Prüfer code gives a unique tree. \square

Corollary A.16. $\theta_n \circ \gamma_n^{-1} : \hat{A}_n \cup \hat{D}_n \rightarrow T_{n+1} \times \{0, 1\}$ is a bijection.

Proof. Directly seen from the lemmas above. \square

Remark. Since \hat{A}_n and \hat{D}_n are mirrors of each other, we know from the above that $\hat{A}_n \cong T_{n+1} \cong \hat{D}_n$, but a direct map is harder to find. Unfortunately, the element of $\{0, 1\}$ does not distinguish ascents vs. descents. There are many trees in T_{n+1} where choosing 0 results in one descent in a parking function, and choosing 1 results in a descent in a different parking function, rather than an ascent.

111 - - - 11 - 1 - - 11 - -1- 1 - 11 - - 1 - 1 - 1-

Fig. 3. Ballot problem for $n = 3$

B. Weighted Ballots

1. The Ballot Problem

There are many representations of the Catalan numbers. In this section, we will examine the ballot problem and we will extend it. This extension will be used when we are counting inversions, as it gives us a method to map to parking functions from permutations. The ballot problem examines the number of ways a series of $2n$ votes can be given such that a given candidate never trails and the voting ends in a tie. This is represented by looking at arrangements of n copies each of 1 and -1 (which we will represent as just $-$), whose partial sums are always non-negative, and whose total sum is zero. (See figure 3 for an example.) From this representation, it is easy to find natural bijections to most other representations of the Catalan numbers. For example, the Dyck paths are given by reading each 1 as an up step and each $-$ as an over step. The formula for the Catalan numbers is well established to be $C_n = \frac{1}{n+1} \binom{2n}{n}$ [25].

A common extension to the Catalan numbers is the p-Catalan numbers [10][9], where p is any positive integer. Under this extension, the usual Catalan numbers are considered the 2-Catalan numbers. For the ballot problem, instead of n positive votes of 1 and n negative votes of -1 , we have n positive votes of $p-1$ and $n(p-1)$ negative votes of -1 . (We retain the partial sum and total sum requirements on the sequences.) For example, for $p = 3$, $n = 2$, the three possible sequences are given by figure 4. Note that this extension is one-sided: that is, it changes the weights on one side (the increases) but not on the other (the decreases). The formula for the

22 - - - - 2 - 2 - - - 2 - -2 - -

Fig. 4. 3-Catalan ballot problem

p -Catalan numbers is $C_{n,p} = \frac{1}{pn+1} \binom{pn+1}{n} [10][9]$.

2. One-sided Generalization

Now consider the case where we allow the weights on the positive side to vary instead of being equal. For example, fix a weight set \mathbb{B} , as a multiset over \mathbb{N} such that the sum of all the elements in \mathbb{B} is equal to n . Note that \mathbb{B} can also be considered a partition of n . Now we can ask how many arrangements exist with the elements of \mathbb{B} as our positive numbers and n copies of -1 that have non-negative partial sums and a total sum of zero. Restricting the multi-set \mathbb{B} to $\{1^n\}$ returns us to the Catalan number case detailed above, and a weight set of $\{(p-1)^n\}$ is the p -Catalan number case.

Theorem B.1. *Assume $\mathbb{B} = \{1^{a_1}, 2^{a_2}, \dots, n^{a_n}\}$, with $\sum a_i = m$, $\sum ia_i = n$. The number of valid ballot arrangements with weighted positive votes given by the multiset \mathbb{B} over \mathbb{N} is given by $\binom{n+1+m}{n+1, a_1, a_2, \dots, a_n} \frac{1}{n+1+m}$.*

Proof. Define $W(\mathbb{B})$ to be the set of words created by taking $n+1$ copies of -1 and the elements of \mathbb{B} , and arranging them in any order. There are $\binom{n+m+1}{n+1, a_1, a_2, \dots, a_n}$ elements of $W(\mathbb{B})$. Now, we set up an equivalence relation among these words, such that any two words are equivalent if one is the cyclic shift of the other.

Given any $v = (v_1 v_2 \dots v_{n+m+1}) \in W(\mathbb{B})$, will we find a member of its equivalence class which has non-negative partial sums excepting the final element (which must be a -1). Let $z = \min\{i : \sum_{j=1}^i v_j < 0 \text{ and } \sum_{j=i+1}^k v_j > 0 \text{ for all } k > i\}$. The set

gives all points where the partial sum of the letters in the word is negative and all partial sums of the following letters are non-negative, and v_z , is the first such point. Thus, the elements after v_z form the maximal "tail" which has non-negative partial sums. If $z = n + m + 1$, let $v' = v$. Otherwise, we move the tail to the front of the sequence, forming $v' = (v_z v_{z+1} \dots v_{m+n+1} v_1 \dots v_{z-1})$. Now, partial sums of v' will be non-negative up to v'_{m+n} , and the last element of v' will be a -1 .

v' is in the same equivalence class as v : Since v' is simply a cyclic shift of v by z , they are in the same equivalence class.

v' is unique for each equivalence class: Assume, to the contrary, that we have two elements v and w that belong to the same equivalence class and $v' \neq w'$. Let x be the cyclic shift of v' to obtain w' , *i.e.*, $w' = (v_x v_{x+1} \dots v_{m+n+1} v_1 \dots v_{x-1})$. Now, by our formation of v' , we know that $\sum_{i=1}^{x-1} v_i$ is non-negative and we know that $\sum_{i=1}^{m+n+1} w_i = -1$. But this means that $\sum_{j=1}^{m+n+1-x} w_j = \sum_{i=x}^{m+n+1} v_i < 0$, which violates the restrictions on w' , giving us a contradiction.

If we remove the final -1 from v' we see that we have a unique element of \mathbb{B} . Since there are $n + m + 1$ numbers in each sequence, there are $n + m + 1$ members of each equivalence class, so we have that the number of valid arrangements of \mathbb{B} is $\binom{n + m + 1}{n + 1, a_1, a_2, \dots, a_n} \frac{1}{n + m + 1}$. \square

The sequences v' that we found and the equivalence relation on the v is a version of Lukasiewicz words, as seen in Stanley[25, Example 6.6.7].

We refer to this generalization as the "left weighted Catalan numbers" since we are only allowing the positive weights to vary, and fixing the negative weights at negative one (The term "Left" comes from the parenthesis representation of the Catalan numbers, where the left parenthesis were weighted; this was the original representation under investigation for use in the Inversion problem given later.)

Central to this problem is the weight set \mathbb{B} , which determines the structure we are looking at, and also acts as a partition of $[n]$. Given a partition, \mathbb{B} , we get the formula above for the number of ballot sequences we have for that partition. If we sum this over all partitions \mathbb{B} of $[n]$, we get the Large Schröder numbers. We can see this by considering Schröder's second problem stated as Plane trees with n leaves and no vertex of degree one by Stanley [25, Example 6.2.8]. In the map given above, the ballot structure maps to trees with no nodes of degree one. So this generalization sits between the Schröder numbers above as their sum and the p -Catalan numbers below as a generalization of them.

We will explore this extension further in chapter IV.

3. Plane Trees of a Given Type

Given a plane tree on $b+1$ nodes, we say that the type of the tree is $(a_0, a_1, a_2, \dots, a_b)$ if there are a_i nodes with i children for all i (also known as the degree of the node). [25, p. 30]

Theorem B.2. *If \mathbb{B} is the multiset $\{1^{a_1}, 2^{a_2}, \dots, (n)^{a_n}\}$, and $n = \sum ia_i$, then the valid ballot arrangements with weighted positive votes given by $\mathbb{B} \cong$ the plane trees of type $(n+1, 0, a_1, a_2, \dots, a_n)$.*

Proof. Order the nodes in the tree by traversing the tree in pre-order (pre-order is parent, left-most child, \dots , right-most child), and assign each node (omitting the final leaf) a weight equal to its degree minus one, giving us weights w_1, \dots, w_{m+n} , where $m = \sum a_i$. Note that the weights record how many new branches of the tree are created by that node as you traverse in pre-order. Thus leaves, which end a branch of the tree, have a weight of -1 , and a node with i children opens up $i-1$ active branches of the tree. Omitting the final leaf, which finishes off the tree, the

partial sums must be non-negative, since you cannot close more branches than you had opened. Thus, a tree with type $(n + 1, 0, a_1, a_2, \dots, a_n)$ will correspond to a sequence $(w_1, w_2 \dots w_{m+n})$ with non-negative partial sums, total sum of zero, and a_i copies of i for all i . The inverse map is clear from the above. (See theorem 5.3.10 and Lemma 4.7.12 in Stanley[25]) \square

If we restrict our multiset \mathbb{B} to $\{1^n\}$ for our trees, we get trees with type $(n + 1, 0, n)$. This is the set of plane binary trees with $2n + 1$ nodes, another well-known representation of the Catalan numbers (See **(d)** in chapter IV). In fact, if we take a multiset of n copies of $p - 1$ for \mathbb{B} , we get the set of plane p -ary trees with $pn + 1$ nodes. Our generalized case, however, allows plane trees with varying degrees on the nodes, so long as none of them is unary.

The figure below shows one tree for $\mathbb{B} = \{1^1, 2^2\}$.

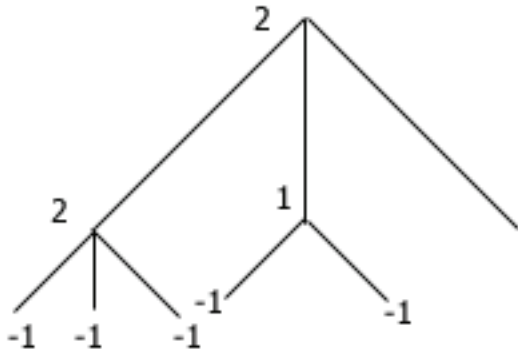


Fig. 5. Tree for $22 - - - 1 - -$

4. Weighted Schröder Ballot Arrangements

Returning to our original ballot problem, let us consider a variation: Instead of only 1 and -1 we also allow 0, so long as the partial sums are non-negative and the total sum is 0. (We refer to these as Schröder ballot sequences, since it is easy to map them to Schröder paths by using the same map we used for Dyck paths and reading a 0 as a diagonal step $(1, 1)$.) Applying the generalization above, we get weighted Schröder ballot sequences.

Theorem B.3. *Given \mathbb{B} and b , the number of valid weighted Schröder ballot arrangements of length $n + b + m$ with b zeroes and positive weights given by \mathbb{B} (whose sum is n and size is m) is $\binom{n + 1 + b + m}{n + 1, b, a_1, a_2, \dots, a_m} \frac{1}{n + b + m + 1}$.*

Proof. We take $[n + b + m]$, and choose b elements to be our zeros. There are $\binom{n + b + m}{b}$ ways to do this. Then we take the remaining elements of $[n + b + m]$, of which there are $n + m$, and assign each either an element of \mathbb{B} or one of the m copies of -1 , requiring that the partial sums be nonnegative. This corresponds to our left weighted ballot problem above, so we know that the number of valid ways to do this is $\frac{1}{n + 1 + m} \binom{n + 1 + m}{n + 1, a_1, \dots, a_m}$. Multiplying the two factors together (and simplifying) gives the desired result. \square

C. Inversions in Parking Functions

Notation and Terminology

Definition C.1. *Similar to the notation for descents, we look at a_i and a_j (where $i < j$) as a transition in the parking function. We call the transition an inversion if $a_i > a_j$, an upversion if $a_i < a_j$, and a long-tie if $a_i = a_j$.*

There are $\binom{n}{2}$ transitions in each parking function of length n . Note that all descents are inversions, all ascents are upversions, and all ties are long-ties (but that the converses do not hold).

Let $PL_{(n,k)} \subset PF_n$ be the set of parking functions of length n with k long-ties.

Let S_n be the set of permutations of length n .

1. Parking Functions and Permutations

Let MB_n be the valid Schröder weighted ballot arrangements of length n . Let S_n be the set of permutations of $[n]$. Given $c = (c_1, \dots, c_n) \in MB_n$, let $M(c) = (m_1, m_2, \dots, m_n)$ be the multiset $\{1^{c_1+1}, \dots, n^{c_n+1}\}$ written down as a sequence in weakly increasing order. Note that, since c has length n and a total sum of zero, $M(c)$ will have exactly n elements. Furthermore, for any valid ballot c , $M(c)$ will be a nondecreasing parking function (or the sorted version of many parking functions).

Given $\pi \in S_n$ and $c \in MB_n$, let $\sigma(\pi, c) = (m_{\pi(1)}, \dots, m_{\pi(n)})$ where $(m_1, \dots, m_n) = M(c)$. In other words, σ permutes the elements of $M(c)$ using the ordering given by π . So σ permutes our sorted parking function and gives us a parking function.

Let $\rho(\pi, c) = (\pi_1, \pi_2, \dots) : S_n \times MB_n \rightarrow \prod_i S_{c_i+1}$ such that ρ gives the orderings of the elements of π for each group of elements in $M(c)$ (in increasing order). Example: If we use $\pi = (12345)$ and $c = (2, -1, 1, -1, -1)$, then $\rho(\pi, c) = (123), (1), (12), (1), (1)$, since there are three copies of 1 and two copies of 3 in our

$M(c) = (11133)$ and the parts of π that correspond to them were in those orders. If we use $\pi = (21354)$ instead, $\rho(\pi, c)$ would be $(213), (1), (21), (1), (1)$ instead.

Given $c \in MB_n$, let PF_c be the set of parking functions which are permutations of $M(c)$ and $\Pi_c = \prod_{c_i \geq 0} S_{c_i+1}$. Let $\tau(\pi, c) = (\sigma, \rho)(\pi, c)$.

Theorem C.2. *For a fixed c in MB_n , $\tau(\pi, c) : S_n \rightarrow PF_c \times \Pi_c$ is a bijection.*

Proof. We will find τ^{-1} : Given f , any parking function in PF_c and $(\pi_1, \pi_2, \dots) \in \Pi_c$, we know that f is just a permutation of $M(c)$. We will recover which permutation generated it from the additional orderings given by the π_i . First, give each element of f a subscript, so that the subscripts form a total ordering of the elements in f , such that $f_i <_! f_j$ if and only if either $[f_i < f_j]$ or $[f_i = f_j \text{ and } f_i \text{ comes before } f_j]$. Then among elements with the same value, but different subscripts, we permute the elements with value i by the inverses of the π_i . The result will be that we can now read the subscripts in the order they appear (ignoring the value of the element they appear on) to recover $\tau^{-1}(f, (\pi_1, \pi_2, \dots))$. \square

For a fixed n , given weight set \mathbb{B} with sum $m \leq n$, let $PF_{\mathbb{B}}$ be the set of parking functions of length n where the multiplicities of the elements minus one are given by the weight set \mathbb{B} , $MB_{\mathbb{B}}$ be the subset of MB_n whose non-negative values are given by \mathbb{B} , and $\Pi_{\mathbb{B}}$ be the union of all Π_c such that $c \in MB_{\mathbb{B}}$.

Corollary C.3. *For a given \mathbb{B} , $\tau(\pi, c) : S_n \times MB_{\mathbb{B}} \rightarrow PF_{\mathbb{B}} \times \Pi_{\mathbb{B}}$ is a bijection.*

Proof. This is the union over all $c \in MB_{\mathbb{B}}$ of the right and left sides above. \square

2. Counting Inversions

Now we will use the above map to count the inversions in parking functions of a given shape, since we know the inversion counts for the other sets in the map. We will do

this by giving generating functions in terms of q where the coefficient of q^i is the number of parking functions in the given set with i inversions.

Theorem C.4. τ preserves inversion counts from S_n to $PF_c \times \Pi_c$.

Proof. Given a c and any $\pi \in S_n$, if π has an inversion from position i to j , then one of two things will be true: Either $M(c)$ has the same element k in position i and j , in which case, the inversion will be carried into the permutation π_k and the elements in the permutation of those positions in the parking function will be the same or $M(c)$ has different elements k_i and k_j in positions i and j , in which case, by the nature of the transformation applied by τ , $i < j$ implies $k_{\pi(i)} < k_{\pi(j)}$ while $\pi(i) > \pi(j)$, which means the inversion will be carried over into the parking function. \square

Definition C.5. For any integer t , we define \mathbf{t}_q to be the polynomial $1 + q + \dots + q^{t-1}$.

We then define the q -factorial, $\mathbf{t}_q!$ to be $\mathbf{t}_q(\mathbf{t}_q - 1) \dots \mathbf{1}_q$. Using the q -factorial, we define the q -multinomial $\binom{\mathbf{s}}{\mathbf{t}_1, \dots, \mathbf{t}_u}_q = \frac{\mathbf{s}_q!}{\mathbf{t}_1! \dots \mathbf{t}_u!}$. [26]

Corollary C.6. The generating function for inversions in parking functions in PF_c is equal to $\binom{\mathbf{n}}{\mathbf{c}_1 + \mathbf{1}, \mathbf{c}_2 + \mathbf{1}, \dots, \mathbf{c}_n + \mathbf{1}}_q$.

Proof. From the previous theorems, we know that the inversions in a permutation of $[n]$ is the same as the number of inversions in the parking function it changes into plus the number of inversions in the permutations of the blocks. The inversions in S_n are known to have generating function $\mathbf{n}_q!$ [26], which means that $\mathbf{n}_q!$ equals the generating function for inversions in PF_c times $\prod_{c_i \geq 0} (\mathbf{c}_i + \mathbf{1})_q$. Division and simplification gives us the result. \square

Corollary C.7. For a fixed \mathbb{B} , the generating function for inversions in parking functions in $PF_{\mathbb{B}}$ is equal to $\binom{\mathbf{n}}{\mathbf{c}_1 + \mathbf{1}, \mathbf{c}_2 + \mathbf{1}, \dots, \mathbf{c}_n + \mathbf{1}}_q \binom{n + 1}{m + 1, \hat{a}, a_1, a_2, \dots, a_m} \frac{1}{n + 1}$,

where the a_i are the multiplicities of the weights in \mathbb{B} , m is the sum of the weights in \mathbb{B} , z is the number of distinct elements in \mathbb{B} , and $\hat{a} = n - z - m$.

Proof. The previous corollary gave a generating function for the number of inversions in parking functions that corresponded to a fixed Schröder weighted ballot arrangement. We can sum over all such arrangements with a specific weight set \mathbb{B} by multiplying by the number of such arrangements, since the generating function does not depend on the specific ballot arrangement chosen, but only on the structure of the weight set \mathbb{B} . \square

CHAPTER III

 K -BLOCKED PARKING FUNCTIONS

A. Introduction

Now we will look at a variant of parking functions, blocked parking functions. First, we hearken back to the original presentation of parking functions by Konheim and Weiss. Imagine a parking lot with n numbered parking spots laid out in a linear fashion, with n drivers wishing to park. Each driver has a preferred spot that they will attempt to park in. If they should find their preferred spot already full, they will move down the line to the first empty spot. If no spot is available at or after their preferred spot, the driver fails to park. An ordered list of parking preferences which allows each driver to park is a valid parking function. By analogy, blocked parking functions are those where some of the spots are listed as already blocked.

Definition A.1. $PB_{(n,k)} = \{(b, a) : b = \{b_1, b_2, \dots, b_k\} \in [n]; a : [n - k] \rightarrow [n]$ represented by $(a_1 a_2 \dots a_{n-k})$ such that if c_1, \dots, c_n is a non-decreasing list of both the a_i and b_i , then $c_i \leq i\}$.

A blocked parking function of length n with k blocked is a set of k blocked spaces followed by an ordered list of $n - k$ parking preferences for the $n - k$ cars. We will refer to the set of parking functions with n spaces with k already blocked as $PB_{(n,k)}$. Clearly, $PB_{(n,0)}$ is PF_n . We present and examine several facts about these blocked parking functions below.

Theorem A.2. *The number of parking functions of length n with k spaces already blocked is $\binom{n+1}{k} (n+1)^{(n-1-k)}$.*

Example: The set $PB_{(2,1)}$ is given by $\{1, [1](2), [2](1)\}$, where the numbers in

the brackets designate the blocked spaces, and the sequence in parenthesis designates the list of parking preferences.

Proof. Recalling the original proof of the size of PF_n , we look at a circular arrangement of $n + 1$ locations numbered from 0 to n . We choose k of them to be blocked and note that set. There are $\binom{n+1}{k}$ such sets that we can generate in this fashion. Now we choose $n - k$ parking preferences. These can be in any space, including the previously blocked ones. There are $(n+1)^{(n-k)}$ such lists of parking preferences. If we then “park” according to the list of preferences after blocking off the list of blocked spots, around the circle, using our collision rule stated above, we will be left with one empty space (call it space f). We rotate both the set of blocked spaces and the parking preferences so that this empty space becomes 0 by subtracting $f \bmod n + 1$ from each value. The result will be a valid parking function, since the method of creating it guarantees that all the values will be between 1 and n (or f would either be a preference for some parker or a blocked spot) and that none of the parkers falls off the end of the lot (or f would have been filled when we placed according to preferences). Now, this method will result in duplicate parking functions for different choices of original blocks and preferences before rotating. However, if we treat as equivalent all choices that result in the same parking function, we note that for each parking function, there are $n + 1$ antecedents, representing the original location of space f before we rotated the circle to place it at 0. This means that the total number of blocked parking functions is $\binom{n+1}{k} \frac{(n+1)^{(n-k)}}{n+1}$, giving the desired result. \square

The formula for this number is highly suggestive. Specifically, we see that

$$\sum_{k=0}^n (n+1-k) \#PB_{(n,k)}$$

$$\begin{aligned}
&= \sum_{k=0}^n (n+1-k) \binom{n+1}{k} (n+1)^{(n-1-k)} \\
&= \sum_{k=0}^n \binom{n}{k} (n+1)^{(n-k)} (1)^k \\
&= (n+2)^n \\
&= \#PF_{n+1}
\end{aligned}$$

This leads us to create a relation between these sets.

Theorem A.3. *Let $I_{(n,k)}$ be the set of (p, b) where p is a blocked parking function in $PB_{(n,k)}$ and b is an element of $[n+1]$ that is not already blocked in p . $I_{(n,k)} \cong PF_n$.*

Proof. We will create a map α which will take a member of $I_{(n,k)}$ to a tree with a root of degree $k+1$. We will do this by creating a Prüfer code of from (p, b) . First, we rotate the values of p by b , (subtracting $b \bmod n+1$ from each element), to get \hat{p} . Since b is not in p 's blocked set, the blocked set of \hat{p} remains non-zero. We treat any zeroes in the parking preferences of \hat{p} as $n+1$. Now we take the set of blocks in \hat{p} and we place zeros in those locations in a Prüfer code. *i.e.*, if \hat{p} 's blocked set is $\{a_1, \dots, a_k\}$, then the Prüfer code (c_1, \dots, c_n) will have $c_{a_i - b \bmod n+1} = 0$ for all i . Then we place the parking preferences from \hat{p} into the code, in order, skipping over space which have already been set to zero. This gives us a Prüfer code with k zeros, which we know translates into a tree with a root of degree $k+1$. For the inverse map, take any tree with $n+2$ nodes and the zero node having degree k , and create the Prüfer code for it. Starting with a circular arrangement of $n+1$ empty spaces, remove the zeroes from the code and place blocks in those locations on the circle. Then use the remainder of the code as parking preferences for $n-k$ cars. This will result in all but one of the spaces on the circle filled. Rotate the circle positively (rotating the blocked set and list of parking preferences) so that this is location zero, recording the size of the rotation

as b . Since this follows our method of formation of blocked parking functions, there will be exactly one rotation which yields a valid blocked parking function.

If we take the union for all k of the $I_{(n,k)}$, we get I_n , and applying α to I_n , we get $LT_{n+2} \cong PF_{n+1}$. This gives us a map that relates every blocked parking function with length n and k spots blocked to a group of $n + 1 - k$ plane trees where the root has degree $k + 1$. \square

B. Blocked Parking Functions and Labeled Schröder Paths

For this section, we will define a labeling on Schröder paths.

Definition B.1. *We will define the “width” of a Schröder path to be the total number of horizontal and diagonal steps, regardless of the “length” of the path which is the number of steps.*

Definition B.2. *Given a Schröder path (using the superdiagonal version of these paths), we label all of the vertical steps with a number from $\{1 \dots n - k\}$, where k is the number of diagonal steps and n is the width of the path. We arrange these labels such that consecutive vertical steps have increasing labels. We will refer to these as labeled Dyck and Schröder paths.*

In B.4, we showed a bijection from the parking functions of length n to labeled Dyck paths of length $2n$. Similarly, let γ be a map from labeled Schröder paths of width n with k diagonal steps to blocked parking functions of length n with k blocks such that each horizontal position i that is blocked in the parking function has a diagonal step at its top (instead of a horizontal step in the path), and that the path otherwise follows the rules for the map from labeled Dyck paths above.

Theorem B.3. *γ is a bijection.*

Proof. Let m be any Schröder path of width n with k diagonals. Because of the super-diagonal requirement on Schröder paths, the number of vertical and diagonal steps before horizontal position i is always greater than or equal to i . Since the vertical steps at position i correspond to parking preferences of i and the (at most one) diagonal step at position i corresponds to a block on i , this means that, if we translate the path into a parking function, we have at least i preferences and blocks less than i (for all $i \leq n$). Thus, the Schröder path satisfies the parking requirement when translated to a parking function. It is straightforward to see that each path defines a unique sorted blocked parking function, and that each labeling of the path (according to our rules above) gives a unique ordering for this set of parking preferences. Therefore, our map is one-to-one. To see that it is onto, we note that every sorted set of parking preferences and blocks yields a Schröder path, and each ordering of that set of parking preferences gives a labeling. \square

Corollary B.4. *The number of labeled Schröder paths (as defined above) with k diagonals of width n is $\binom{n+1}{k}(n+1)^{(n-1-k)}$*

Proof. This is a combination of the theorem above and the count we have already found for blocked parking functions of length n with k blocks. \square

C. p -Parking Functions

Consider a parking function of pairs, (a_i, c_i) where a_i is the parking preference and c_i is a color, chosen from a set of p colors. If we use the same restriction on parking preferences as before, but allow the colors to be arbitrarily chosen, we get a new set of functions. For a given n, p , we see that we can take the previous set of parking functions PF_n and arbitrarily color each parking preference. Thus, if PF_n^p is the set of parking functions colored this way, $\#PF_n^p = \#PF_n p^n = (n+1)^{n-1} p^n$.

Theorem C.1. $PF_n^p \cong LT_{n+1}^p$, the set of labeled trees with $n + 1$ nodes and the edges colored arbitrarily from a set of p colors.

Proof. Given a colored parking function $a = ((a_1, c_1) (a_2, c_2) \dots (a_n, c_n)) \in PF_n^p$, we create the Prüfer code in the usual way for a and assemble the tree from the parking function. However, as we add each node for the transition $a_i a_{i+1}$, we color it with c_i , with the first color c_n going to the edge from the first node added to the 0 node, and the last color c_1 going to the edge for the final node added to the tree. \square

These parking functions and several properties about them were investigated by Stanley previously[23] with a slightly different notation, which involved changing the size restriction on the preferences to be less than or equal to pi . If we take our pairs (a_i, c_i) , we will note that the list of preferences whose i th element is given by $(a_i(p-1)+c_i+1)$ fits this restriction, giving us an easy map between our representation here and the previous one in [23]. (A note: The term used by Stanley for these structures was k -parking functions. However, this term has been used by Yan and Stanley to refer to a slightly different version of parking functions, which require the b_i (the non-decreasing arrangement of the a_i) to be $\leq (i-1)k+1$, which requires that one of the a_i have a value of 1 instead of requiring that at least one of them be $\leq k$. Thus, we have renamed the color version above to p -parking functions to avoid confusion.)

D. p -Parking Functions With k Blocked

We can also consider the case of blocked parking functions when we add this color component to them. We block of spaces as before, and we assign a color to each block in the block set. (If we use the previous notation from[23], this corresponds to blocking p spaces at a time, always ending the blocked section at a space divisible by

p , and denoting a “placement” in the block of one of the p spaces, corresponding to its color here.) When we do this, we see that the previous map to a family of trees with zero node of degree $k + 1$ is unchanged and that our map from p -parking functions to a colorization of these trees can be modified to just require an extra color preference.. This leaves us with a map that takes a k -blocked p -parking function β , along with a rotation r that isn't in the blocked set of β and an extra color c for that rotation, and gives us a labeled tree on $n + 2$ nodes with the edges colored with p colors.

E. Linear Probes in $PB_{(n,k)}$

Definition E.1. Given a parking function $a = [b_1, \dots, b_k](a_1 \dots a_{n-k}) \in PB_{(n,k)}$, we define the number of linear probes in the parking function to be $\alpha(a) = \binom{n+1}{2} -$

$$\sum_{i=1}^k b_i - \sum_{i=1}^{n-k} a_i.$$

If we return to our original description of parking functions, this is the number of occupied spaces that the parking cars try (and fail) to park in from their preference until they reach their final destination.

Theorem E.2. If $f_{n,k}(q) = \sum_{a \in PB_{(n,k)}} q^{\alpha(a)}$, and by convention we set $f_{0,0}(q) = 1$, then

$$f_{1,1}(q) = f_{1,0}(q) = 1, \text{ and } f_{n+1,k}(q) = \sum_{j=0}^k \sum_{i=j}^{n-(k-j)} \binom{n-k}{i-j} (\mathbf{i} + \mathbf{1})_{\mathbf{q}} f_{i,j}(q) f_{n-i,k-j}(q).$$

Proof. Let $a = [b_1 \dots b_k](x_1 \dots x_{n+1-k}) \in PB_{(n+1,k)}$. Let a_m be the largest value that x_{n+1-k} could be without violating the constraints of parking functions. We will decompose a into two smaller parking functions by splitting based on whether each element is smaller or larger than a_m . Let $a_s = [b_1 \dots b_j](y_1 \dots y_{i-j})$ where $i = a_m - 1$, $j = \#\{b_r : b_r < a_m\}$, and y_t is the t th element of a that is less than a_m . Similarly, let $a_L = [b_{j+1} - a_m \dots b_k - a_m](z_1 - a_m \dots z_{n-k-i} - a_m)$ where z_t is the t th element

of a that is greater than a_m . Now, $a_s \in PB_{i,j}$ and $a_L \in PB_{n-i,k-j}$, and $\alpha(a) = \alpha(a_s) + \alpha(a_L) + a_m - x_{n-k}$. Now, for each a_s, a_L , and a_m , there are $\binom{n-k}{i-j}$ choices for which $i-j$ of the x belong to a_s , which does not change $\alpha(a)$ and there are $a_m = i+1$ choices for x_{n-k} . Thus, $f_{n+1,k}(q) = \sum_{a \in PB_{(n+1,k)}} q^{\alpha(a)} = \sum_{a_s, a_L, a_m} q^{\alpha(a_s) + \alpha(a_L)} (q^0 + \dots + q^{a_m}) \binom{n-k}{i-j} = \sum_{j=0}^k \sum_{i=j}^{n-(k-j)} \binom{n-k}{i-j} (\mathbf{i} + \mathbf{1})_{\mathbf{q}} f_{i,j}(q) f_{n-i,k-j}(q)$. \square

Remark. Kreweras proved this result for standard parking functions[14] (he used an equivalent structure called major sequences), and the proof above is a generalization of his method for blocked parking functions, which yielded the following result:

Corollary E.3. If $P_n(q) = \sum_{a \in PF_n} q^{\alpha(a)}$, and by convention we set $P_0(q) = 1$, then $P_1(q) = 1$ and $P_{n+1}(q) = \sum_{i=0}^n \binom{n}{i} P_i(q) P_{n-i}(q) (\mathbf{i} + \mathbf{1})_{\mathbf{q}}$.

Proof. This can be seen by fixing $k = 0$ in E.2. \square

Beissinger and Peled found a map from linear probes to external activity in trees and thus to inversions in trees[1].

The formula in E.2 is more complex than is desirable. What follows is an examination of the formulas for $PB_{n,n-j}$ for a fixed j while varying n .

Theorem E.4. $f_{n,n}(q) = 1$, and thus the generating function for it is $\sum_{n>0} f_{n,n}(q) y^n = \frac{y}{1-y}$

Proof. If we require n blocks and a length of n , the only case is $[1, \dots, n]$. This gives us the formula for $f_{n,n}$. Thus, the sum gives us $y + y^2 \dots = \frac{y}{1-y}$. \square

Theorem E.5. $f_{n,n-1}(q) = \sum_{j=0}^n (n-j) q^j$.

Proof. We have a length of n and $n - 1$ blocked, giving us one unblocked space. If we have j linear probes, the distance between the unblocked space and the single parking preference must be j . There are $n - j$ possible choices of a_1 such that $1 \leq a_1 - j < a_1 \leq n$. \square

Corollary E.6. *The generating function* $\sum_{n>0} f_{n,n-1}(q)y^n = \frac{y}{(1-y)^2(1-xy)}$.

Proof. Letting $k = n - j$ in the above theorem, we have

$$\begin{aligned}
& \sum_{k>0} \sum_{j>0} ky^{k+j}q^j \\
&= \sum_{j>0} q^j y^j \sum_{k>0} ky^k \\
&= \sum_{j>0} q^j y^j \sum_{k>0} y \frac{\delta}{\delta y} y^k \\
&= \sum_{j>0} q^j y^{j+1} \frac{\delta}{\delta y} \sum_{k>0} y^k \\
&= \sum_{j>0} q^j y^{j+1} \frac{\delta}{\delta y} \frac{1}{1-y} \\
&= \sum_{j>0} q^j y^{j+1} \frac{1}{(1-y)^2} \\
&= \frac{y}{(1-y)^2} \sum_{j>0} q^j y^j \\
&= \frac{y}{(1-y)^2(1-xy)}
\end{aligned}$$

\square

We will now give a formula and generating function for $f_{n,n-2}(q)$, but this will require some setup. Given a blocked parking function $a = [b_1, \dots, b_k](a_1 \dots a_{n-k}) \in PB_{(n,k)}$, we will say that a is reducible if $b_1 = 1$ and $a' = [b_2 - 1, \dots, b_k - 1](a_1 \dots a_{n-k})$ is a valid blocked parking function in $PB_{(n-1,k-1)}$. We will call a' the reduction of a .

Theorem E.7. $\{a \in PB_{(n+1,n-1)} \mid a \text{ is reducible}\} \cong PB_{(n,n-2)}$.

Proof. Clearly, reducing is a one-to-one operation. In order to see that it is also onto, we take any blocked parking function $a' = [b_1, \dots, b_{n-2}](a_1 a_2) \in PB_{(n,n-2)}$ and we create $a = [1, b_1 + 1, \dots, b_{n-2} + 1](a_1 a_2)$. There are $n - 1$ blocks in a . We let $c_1 \dots c_{n+1}$ be the blocks and preferences of a sorted into non-decreasing order. $c_1 = 1$, because we inserted a 1 at the beginning of the blocks. If a is not a valid blocked parking function, fix k to be the smallest integer such that $c_k > k$. Now, if the offending element is a block, $c_k - 1 > k - 1$, and a' was not a valid blocked parking function. If the offending element was a parking preference, $c_k > k - 1$, and a' was not a valid blocked parking function. Therefore, there can be no such k , so a is a valid blocked parking function. Thus, the operation above is onto as well as one-to-one, giving us a bijection. \square

Corollary E.8.
$$\sum_{\substack{a \in PB_{(n+1,n-1)} \\ a \text{ is reducible}}} q^{\alpha(a)} = q^2 f_{n,n-2}.$$

Proof. First, note that $\alpha(a) = \alpha(a') + 2$, since we have added one linear probe to each of a_1 and a_2 by shifting the blocks up by one and adding a block in the new first space. Then the bijection above gives us the required formula \square

Theorem E.9. $\#\{a \in PB_{(n+1,n-1)} : \alpha(a) = 0\} = n(n + 1)$.

Proof. If we have no linear probes and all but two spaces are blocked, we know that the parking preferences must be the unblocked spaces. We choose a_1 and a_2 from the $n + 1$ possible choices, and the blocked spaces are automatic. Noting that both orderings for the choices give a distinct blocked parking function, we have $2 \binom{n + 1}{2} = n(n + 1)$ possible blocked parking functions with 2 unblocked spaces and no linear probes. \square

We give the following corollary to highlight that the two sets of blocked parking functions discussed so far are disjoint.

Corollary E.10. *Blocked parking functions without linear probes are not reducible.*

Proof. While it is possible to prove this directly, we note that the q -polynomial for the reducible blocked parking functions has smallest term q^2 , leaving no constant term, which means there are no reducible parking functions with fewer than two linear probes. \square

We will now count the remaining blocked parking functions, that is, blocked parking functions with at least one linear probe that are not reducible.

Theorem E.11. *The number of non-reducible blocked parking functions in $PB_{(n+1,n-1)}$ with i linear probes is $(n-i)(2(n-i)-1)$.*

Proof. Let $a = [b_1, \dots, b_{n-1}](a_1 a_2)$ be non-reducible. Let g_1 and g_2 be the unblocked spaces (*i.e.*, the elements missing from the b_i). Since a is non-reducible, we know that either $b_1 \neq 1$ or a' is not a valid blocked parking function. In the first case, we know that one of our gaps (and therefore, one of our a_i) must be equal to 1, and the other gap must be i larger than the other a_i . In the second case, let c_i be the preferences and blocks of a sorted in a non-decreasing order. We know that $c_i \leq i$. Let c'_{i-1} be the sorted preferences and blocks of a' . Let j be the smallest integer such that $c'_{j-1} > j-1$ (*i.e.*, this is the element that makes a' an invalid blocked parking function). However, since a is valid, we know that c_j is a parking preference, $c_j = c'_j$ and $c_j \leq j$. This means that $c_j = j$. Furthermore, we know that $c_{j-1} < c_j$, so c_j is also the location of a gap. This means that we again have one preference equal to one of the gaps, which means the other preference must be i less than the corresponding gap in the blocks.

Thus, in both cases, we have to choose the gaps and the preferences, but we know that one of the gaps equals a preference, and the other gap is equal to the other preference plus i . Now, if the preferences are the same, this gives us $n-i$

choices for $a_1 = a_2$, and one of the gaps equals this number, and the other is i larger. If the preferences are different, we can choose any of the $n - i$ upper elements for our first gap, and any element not between that gap and its corresponding parking preference (which is i lower than it) for our second element. Since both orderings for the a_i produce distinct parking functions, we have $2(n - i)(n - i - 1)$ choices when we require the a_i to be distinct and $n - i$ choices when we allow them to be identical.

Adding the two together gives us the total number of non-reducible blocked parking functions with i linear probes as $(n - i)(2(n - i) - 1)$. \square

Corollary E.12. $f_{n+1,n-1}(q) = n(n + 1) + \sum_{i=1}^n [(n - i)(2(n - i) - 1)q^i] + q^2 f_{n,n-2}(q)$

Proof. This corollary is the sum of the formulas for the disjoint sets from E.8, E.9, and E.11. \square

We will now give a series of lemmas we will use to find a generating function for the function $f_{n+1,n-1}(q)$.

Lemma E.13. $\sum_{n>0} n(n + 1)y^{n+1} = \frac{2y^2}{(1 - y)^3}$.

Proof. We manipulate $\sum_{n>0} n(n + 1)y^{n+1}$

$$\begin{aligned} &= \sum_{n>0} y^2 \frac{\delta^2}{\delta y^2} y^{n+1} \\ &= y^2 \frac{\delta^2}{\delta y^2} \sum_{n>0} y^{n+1} \\ &= y^2 \frac{\delta^2}{\delta y^2} \frac{y^2}{1 - y} \\ &= \frac{2y^2}{(1 - y)^3} \end{aligned}$$

\square

Lemma E.14.
$$\sum_{n>0} y^n \sum_{i=1}^n i q^{n-i} = \frac{-qy}{(1-y)(1-yq)(1-q)}.$$

Proof. We manipulate
$$\begin{aligned} \sum_{n>0} y^n \sum_{i=1}^n i q^{n-i} &= \sum_{n>0} y^n q^n \left[\sum_{i=1}^n i p^i \right]_{p=\frac{1}{q}} \\ &= \sum_{n>0} y^n q^n \left[\frac{p(1-p^n)}{(1-p)^2} \right]_{p=\frac{1}{q}} \\ &= \sum_{n>0} y^n \frac{q^n (1 - \frac{1}{q^n})}{q(1 - \frac{1}{q})^2} \\ &= \sum_{n>0} y^n \frac{q(q^n - 1)}{(1-q)^2} \\ &= \sum_{n>0} y^n q^n \frac{q}{(1-q)^2} - \sum_{n>0} y^n \frac{q}{(1-q)^2} \\ &= \frac{q}{(1-q)^2} \left(\frac{1}{1-yq} - \frac{1}{1-y} \right) \\ &= \frac{-qy}{(1-y)(1-yq)(1-q)} \end{aligned}$$

□

Lemma E.15.
$$\sum_{n>0} y^n \sum_{i=1}^n i^2 q^{n-i} = \frac{1+q}{(1-q)^2(1-yq)(1-y)}.$$

Proof. We manipulate
$$\begin{aligned} \sum_{n>0} y^n \sum_{i=1}^n i^2 q^{n-i} &= \sum_{n>0} y^n q^n \left[\sum_{i=1}^n i^2 p^i \right]_{p=\frac{1}{q}} \\ &= \sum_{n>0} y^n q^n \left[(1-p^{n+1}) \sum_{i>0} i^2 p^i \right]_{p=\frac{1}{q}} \\ &= \sum_{n>0} y^n q^n \left[(1-p^{n+1}) \frac{p(p+1)}{(1-p)^3} \right]_{p=\frac{1}{q}} \end{aligned}$$

$$\begin{aligned}
&= \sum_{n>0} y^n \frac{q^n \left(1 - \frac{1}{q^{n+1}}\right) \left(\frac{1}{q} + 1\right)}{q \left(1 - \frac{1}{q}\right)^3} \\
&= \sum_{n>0} y^n \frac{(q^{n+1} - 1)(1 + q)}{(q - 1)^3} \\
&= \frac{1 + q}{(q - 1)^3} \left(q \sum_{n>0} y^n q^n + \sum_{n>0} y^n \right) \\
&= \frac{1 + q}{(q - 1)^3} \left(\frac{q(1 - y) - (1 - yq)}{(1 - yq)(1 - y)} \right) \\
&= \frac{1 + q}{(1 - q)^2(1 - yq)(1 - y)}
\end{aligned}$$

□

Corollary E.16. $\sum_{n>0} f_{n+1,n-1}(q)y^n = \frac{1}{1 - q^2} \left(\frac{2y^2}{(1 - y)^3} + \frac{2 + 2q + qy - q^2y}{(1 - q)^2(1 - yq)(1 - y)} \right).$

Proof. Let $\mathcal{F}(y, q) = \sum_{n>0} f_{n+1,n-1}(q)y^n$. By E.12, this is

$$\begin{aligned}
&= \sum_{n>0} y^n \left[n(n + 1) + \left(\sum_{i=1}^n i(2i - 1)q^{n-i} \right) + q^2 f_{n,n-2}(q) \right] \\
&= \sum_{n>0} y^n \left[n(n + 1) + 2 \left(\sum_{i=1}^n i^2 q^{n-i} \right) - \left(\sum_{i=1}^n i q^{n-i} \right) + q^2 f_{n,n-2}(q) \right]
\end{aligned}$$

Using E.13, E.14, and E.15, we get

$$\mathcal{F}(y, q) = \frac{2y^2}{(1 - y)^3} + \frac{2(1 + q)}{(1 - q)^2(1 - yq)(1 - y)} + \frac{qy}{(1 - q)(1 - yq)(1 - y)} + q^2 \mathcal{F}(y, q)$$

Moving the last term to the left side and dividing the whole by $1 - q^2$ gives us the result. □

Remark. If we examine $f_{n+1,n+1-i}$, we see that we will still have the reducible parking functions giving us a term of $q^i f_{n,n-i}$ and the parking functions with no linear probes giving us a term of $\frac{(n + 1)!}{(n + 1 - i)!}$. The remaining terms will come from the non-reducible parking functions with at least one linear probe. A general formula for

these terms is an avenue for further exploration.

CHAPTER IV

LEFT WEIGHTED CATALAN STRUCTURES

In this chapter, we explore various structures enumerated by the Catalan and p -Catalan numbers and apply the generalization to left weighted Catalan numbers as developed in II.B, giving us an extension of these structures to include our weight set \mathbb{B} .

We will make frequent references to Stanley's list of Catalan representations, from Enumerative Combinatorics II[25][27]. These can be found in exercise 6.19, where each of the representations discussed is given as a part of the exercise. Throughout, we will simply refer to the part of the exercise, such as **(r)** for exercise 6.19.r, which details the ballot problem.

In each of the cases below, we will give the basic Catalan or p -Catalan case, describe the generalized version, and outline a bijection to a previous case. The representations are organized by structure type. Where we refer to a proof for the standard Catalan case, we are referring to the one implied or given by Stanley, except as noted otherwise. In our extension, we will use the following notation: $\mathbb{B} = \{1^{a_1}, 2^{a_2}, \dots, n^{a_n}\}$ is our weight set, $m = \sum_{i=1}^n a_i$ is the cardinality of \mathbb{B} and $n = \sum_{i=1}^n ia_i$. As previously mentioned, \mathbb{B} also functions as a partition on n .

For each Catalan version, we will list the base 2-Catalan case, then give a description of the left weighted Catalan case. For most items, we will just give a sketch of the bijection, rather than a full proof. Finally, we will give five examples from each Catalan representation with the weight set $\mathbb{B} = \{1^1, 2^2\}$ (the same five elements each time). We start with a complete list of the weighted ballots for our \mathbb{B} (figure 6); the bolded elements are those we will be repeating for each subsequent Catalan example.

221 - - - - -	2 - 2 - - 1 - -	2 - - 21 - - -	2 - - 1 - 2 - -
1 - 2 - 2 - - -	12 - - 2 - - -	1 - 22 - - - -	12 - - - - 2 - -
12 - 2 - - - - -	122 - - - - -	2 - - 12 - - -	2 - - 2 - - 1 -
2 - - 2 - 1 - - -	2 - 1 - - 2 - - -	2 - 1 - 2 - - -	2 - 12 - - - - -
2 - 2 - - - - 1 -	2 - 2 - 1 - - -	2 - 21 - - - - -	21 - - - - 2 - -
21 - - 2 - - -	21 - 2 - - - - -	212 - - - - -	22 - - - - 1 -
22 - - - - 1 - -	22 - - 1 - - -	22 - 1 - - - - -	

Fig. 6. Weighted ballots (Catalan case (\mathbf{r})) $\mathbb{B} = \{1^1, 2^2\}$

A. Trees

(c) Catalan: The set of binary trees with n nodes.

Generalization (figure 7): We look at binary trees with n nodes, such that the tree is partitioned into “straight line” groups of nodes, such that the number of nodes in the groups is given by \mathbb{B} (*i.e.*, there are a_i groups of size i , for all i). To make a straight line group of nodes, take a node, called the base, and note which child of its parent it is. Then take that child of the base, and that child of the child, etc., until we have i nodes in a straight line. (Define the root of the tree to be a left child). This reduces to the Catalan case where each node is its own group, *i.e.*, we always use the trivial partition of singletons.

We can create a bijection to (\mathbf{r}) by starting with the root, counting the number of nodes in its straight line group as the first positive number. Then we examine the children of this group, in pre-order, and record a $-$ for each missing child. Once a new group is found, repeat the process on the new group. (As usual, omit the final -1). This will give us a sequence of positive weights counting the nodes in groups, and the list of -1 can never exceed the number of nodes, since each node only adds

one additional possibility for a -1 (until the final missing child, which we omit).

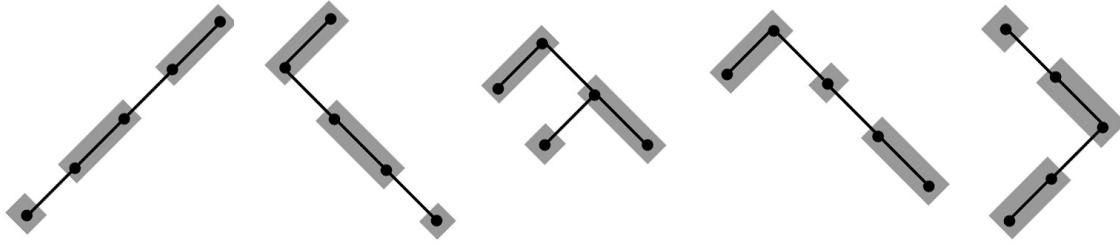


Fig. 7. Weighted Catalan examples for (c)

(d) Catalan: Plane trees with n internal nodes, all of degree 2. (Each node has 0 or 2 children).

p -Catalan: Plane trees with n internal nodes, all of degree p . (Each node has 0 or p children.)

Generalization (figure 8): Trees of type $(n + 1, 0, a_1, a_2, \dots, a_n)$. (See theorem B.2 for bijection.)

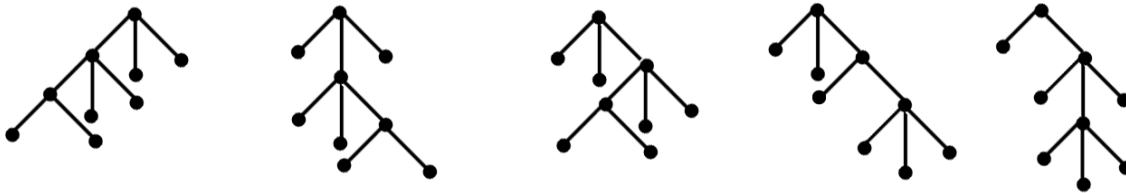


Fig. 8. Weighted Catalan examples for (d)

(e) Catalan: The set of plane trees with $n + 1$ nodes.

Generalization (figure 9): we look at plane trees with $n + 1$ nodes with a partition on the edges of the tree such that the sizes of the blocks are given by \mathbb{B} and in each block, all but the topmost edge is the leftmost child of its root. This reduces to the Catalan case where each edge is its own block, *i.e.*, we always use the trivial partition of singletons.

In the Catalan case, we create a bijection to **(r)** by reading the tree in pre-order, and treat each step downward as a 1 and each step back up as a -1 . In the generalized case, we do the same, but read the entire block of a partition as a single positive number.

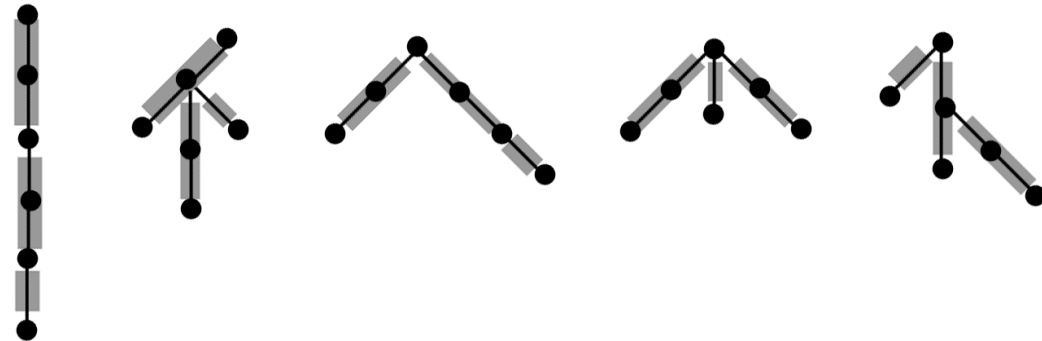


Fig. 9. Weighted Catalan examples for **(e)**

(f) Catalan: Planted trivalent trees with $2n + 2$ nodes.

Generalization (figure 10): Planted trees with with internal node valences given by $v_2 = 0$ and for $i > 2$, v_i is the number of $i - 2$ in \mathbb{B} (and the root has valence 1). In the Catalan case, the internal valences given are of size 3, corresponding to the n copies of 1 in \mathbb{B} .

As in the Catalan case, we find a bijection to **(d)** by cutting the root to get a tree of type $(n + 1, 0, v_3, \dots)$.

(g) Catalan: Plane trees with $n + 2$ nodes such that the rightmost path of each subtree of the root has even length.

Generalization (figure 11): Plane trees as in **(e)** (with an extra group of size 1 added, to make $n + 2$ nodes) with the restrictions that the rightmost path of any subtree of the root is required to be of even length, and the leftmost child of the root will be its own block in the partition.

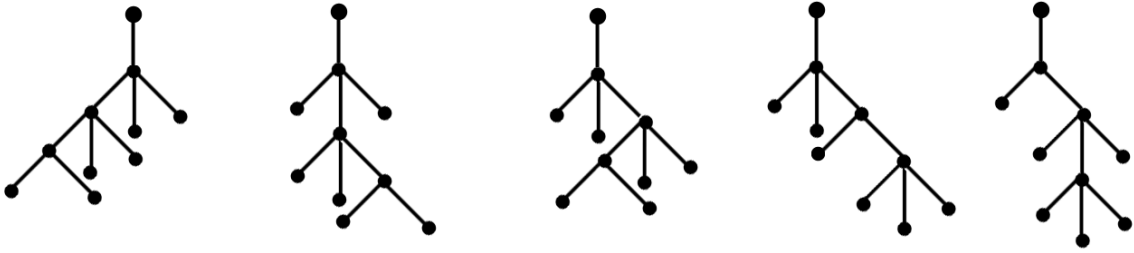


Fig. 10. Weighted Calatan examples for (f)

The bijection to (e) is the same as in the Catalan case: Examine the tree and find the rightmost subtree of the root which has an odd rightmost path. Insert a new node as the leftmost child of the root and move that subtree and every subtree to its left to be a child of this new node. (For the partition, treat this new node as a singleton.) Then the rightmost path of the subtree of this new edge will be of even length. The rest of the features of (e) will be maintained. If there is no such subtree, insert the new edge as the leftmost subtree of the root. To convert back to (e), simply remove the leftmost edge of a tree, which will, by construction, be the only node in its block in the partition of the nodes.

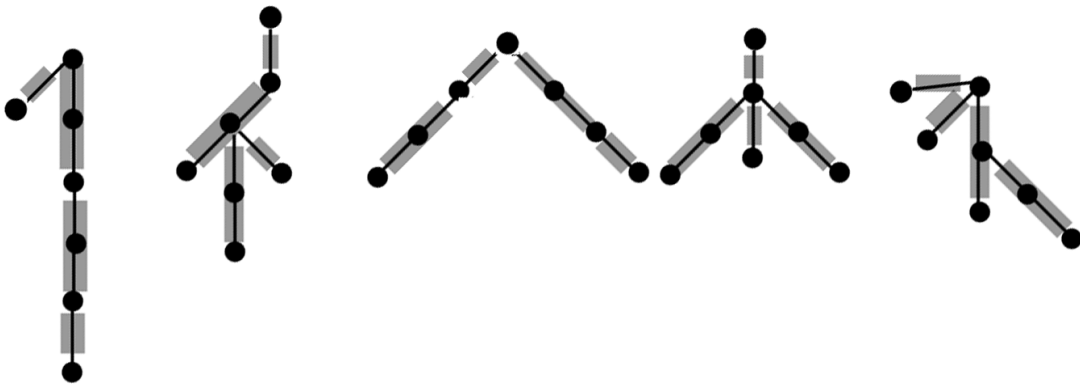


Fig. 11. Weighted Calatan examples for (g)

B. Lattice Sequences

In standard terminology, a lattice path is a path traced out on a grid between points (x, y) such that x and y are positive integers and each step along the path is of unit length. For example, one definition of the Dyck paths is that they are lattice paths traced out by steps $(0, 1)$ and $(1, 0)$ from $(0, 0)$ to (n, n) and never fall below the diagonal $x = y$. In this section, we use the term sequence, as opposed to path, to indicate that we want to distinguish the orderings of the sizes of the steps, and not just the paths they trace out. In other words, we treat the sequence $(0, 2), (0, 1)$ as distinct from $(0, 1), (0, 2)$, even though they trace out the same path. When drawing these sequences, it is helpful to think of each step as an edge, and having nodes at the end points of each step. Thus the steps given would result in two graphs of a line segment from $(0, 0)$ to $(0, 3)$, but the former would have a node at $(0, 2)$ and the latter at $(0, 1)$, both marking the end of the first step of their respective graphs.

(h) p -Catalan: Lattice paths from $(0, 0)$ to (n, n) with steps $(0, p - 1)$ or $(1, 0)$, never rising above the line $y = x$.

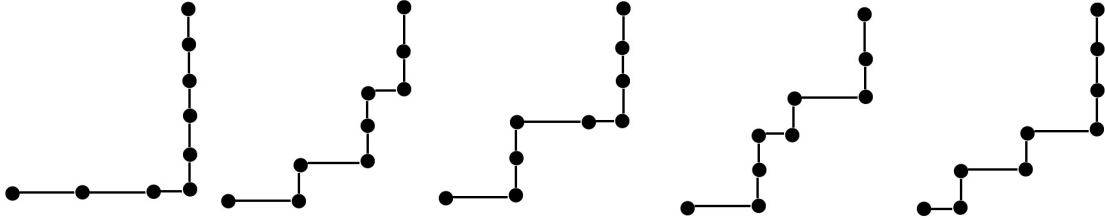
Generalization (figure 12): Lattice sequences from $(0, 0)$ to (n, n) with steps $(0, k)$ or $(1, 0)$, never rising above the line $y = x$, where the multiset of k is given by \mathbb{B} .

As in the Catalan case, we form a bijection with (\mathbf{r}) by reading each horizontal step of size k as the positive integer k , and each vertical step as a -1 . The restriction on the diagonal corresponds to the non-negativity of the sequences for (\mathbf{r}) .

(i) Catalan: Paths from $(0, 0)$ to $(2n, 0)$ with steps $(1, -1)$ and $(1, 1)$, never falling below the x-axis.

Generalization (figure 13): Sequences from $(0, 0)$ to $(2n, 0)$ with steps $(1, -1)$ and (k, k) , never falling below the x-axes, where the multiset of the k is given by \mathbb{B} .

As in the Catalan case, this is a rotation and rescaling of (\mathbf{h}) .

Fig. 12. Weighted Catalan examples for **(h)**Fig. 13. Weighted Catalan examples for **(i)**

(j) Catalan: Paths from $(0, 0)$ to $(2(n+1), 0)$ with steps $(1, -1)$ and $(1, 1)$, never falling below the x -axis, such that any maximal sequence of consecutive $(1, -1)$ steps ending on the x -axis had odd length.

Generalization (figure 14): Paths from $(0, 0)$ to $(2(n+1), 0)$ with steps $(1, -1)$ and (k, k) , never falling below the x -axis, such that any maximal sequence of consecutive $(1, -1)$ steps ending on the x -axis had odd length, where the multiset of the k is given by \mathbb{B} with an additional element of size 1, which will always be the first step.

As in the Catalan case, take any sequence from **(i)** and locate the rightmost descent of even length onto the x -axis. Insert a $(1, -1)$ step into this descent and a $(1, 1)$ step at the beginning of the sequence. If there is no such descent, just insert a $(1, 1)(1, -1)$ sequence at the beginning. This will raise the rightmost even descent to an odd one, and all previous descents (even or not) will no longer touch the x -axis. The inverse bijection is to remove the first $(1, 1)$ from a given sequence and the final

$(1, -1)$ from the leftmost descent that touches the x-axis.

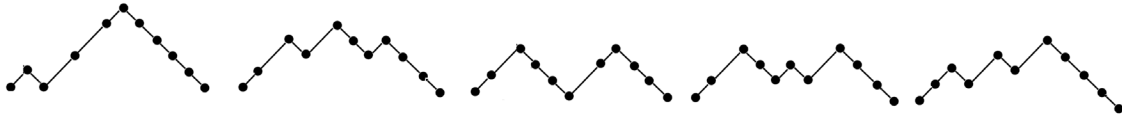


Fig. 14. Weighted Catalan examples for **(j)**

(k) Catalan: Sequences from $(0, 0)$ to $(2(n + 1), 0)$ with steps $(1, -1)$ and $(1, 1)$, never falling below the x-axis, with no peaks at height 2.

Generalization (figure 15): The same sequences but with steps $(1, -1)$ and (k, k) where the multiset of k are given by \mathbb{B} plus an additional element of size 1.

The bijection for this follows that of the Catalan case[20]. Start with a sequence from **(i)**, and add a single up and down step to the front. For each maximal subsequence of our sequence containing no peaks at height one, raise the subsequence by one by adding an up step at the front and a down step at the back. Now, we have added i pairs of steps, 1 for each such maximal subsequence. However, each such subsequence must be preceded by a peak of height one. Remove the peak of height one immediately preceding each such subsequence to remove i pairs of steps, leaving us with a sequence with $n + 1$ pairs of steps and no peaks at height 2.



Fig. 15. Weighted Catalan examples for **(k)**

(l) Catalan: Noncrossing pairs of sequences of $n + 1$ steps $(1, 0)$ and $(0, 1)$, which only intersect at start and end.

Generalization (figure 16): Pairs of sequences of steps of length $n + 1$, such that

the upper sequence has positive steps of $(0, k)$ and negative steps of $(1, 0)$, the lower sequence has positive steps of $(k, 0)$ and negative steps of $(0, 1)$, the sequences only join at beginning and end, the bottom sequence starts with an additional step of $(1, 0)$, and the multiset of the k (other than the initial step of the bottom sequence) is given by \mathbb{B} .

This is in bijection to (\mathbf{r}) . Read each path, alternating from the upper to the lower (skipping the first step of the lower), and record positive steps as positive integers of the size of the step, and negative steps as -1 , ignoring the final element. By alternating, in this context, we mean that when the total size of the steps read from one path so far equals or exceeds the total number of steps read from the other path so far, switch paths (ignoring the extra step that starts the lower path to maintain separation). The partial sums of the sequence in (\mathbf{r}) plus one will correspond to the separation between the paths in grid steps. Until the final element, the separation will always be at least one, corresponding to a non-negative partial sum.

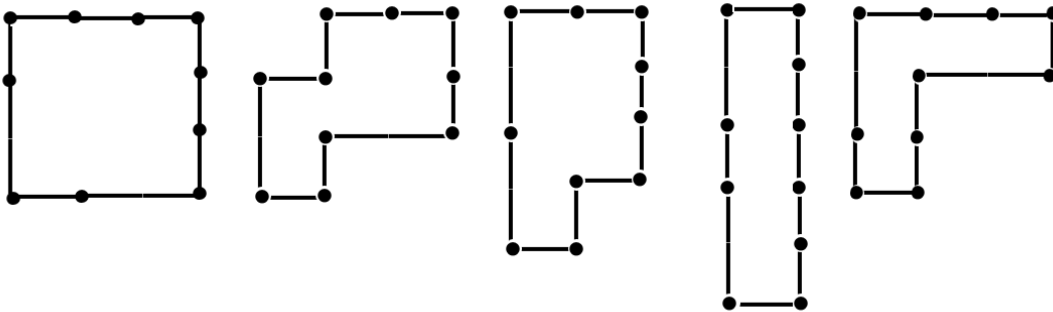


Fig. 16. Weighted Calatan examples for $(\mathbf{1})$

C. Partitions

Several of these representations distill down to pairs of partitions where the second partition has block sizes given by \mathbb{B} and the first partition is some less refined partition. These are intervals on the poset of partitions of $[n]$, and the set of such intervals is the set of intervals on the partition poset of $[n]$ whose lower bounds have block sizes given by \mathbb{B} . Stanley[23] proved that the labels of the maximal chains of non-crossing partitions with k blocks of $n+1$ are the parking functions of length n , so each interval is actually a set of subsequences of parking functions where the subsequences have the same starting and ending point in the poset, and the lower point of the intervals is being determined by the block sizes in \mathbb{B} .

(\mathbf{r}^*) : We've already discussed (\mathbf{r}) at length above, but we wish to note a slight reinterpretation of it here. If we replace the $+1$ elements with left parenthesis, and the -1 elements with right parenthesis, we get a set of valid parenthesizations, where each closing parenthesis matches to exactly one open parenthesis that preceded it and visa versa.

The $+1/-1$ representation of (\mathbf{r}) corresponds to a non-nesting partition of $2n$ into blocks of size two, which can be seen by connecting the first $+1$ with the first -1 , the second pair, the third pair, etc., up to the n th pair. However, the parenthesis representation (referred to as (\mathbf{r}^*) henceforth) corresponds to non-crossing partitions, since an open parenthesis matches to a close parenthesis such that the two enclose all intermediate parenthesis. This correspondence of non-crossing and non-nesting is well known; these two representations are simply an easy way to demonstrate their equivalence for the rest of the representations.

Generalization (figure 17): Instead of only having open and close parenthesis, use weighted left parenthesis that match up to multiple right parenthesis. This gives

non-crossing partitions of $[2n]$ such that \mathbb{B} gives the block sizes. We represent a weighted right parenthesis by the weight, such as $31))))$ being an example of an opening parenthesis set of weights 3 and 1.

221)))) 2)2))1)) 2))21))) 2))1)2)) 1)2)2))))

Fig. 17. Weighted Catalan examples for (\mathbf{r}^*)

(o) Catalan: In Stanley, this is described as non-intersecting arcs joining n pairs of points in the plane. Our preferred version of this representation is to think of it as partitions of $2n$ into blocks of size 2.

Generalization (figure 18): Non-crossing partitions of $n + m$ where the elements of \mathbb{B} correspond to one less than the sizes of the blocks in the partition. In terms of the original representation, this would entail nonintersecting arcs joining $2n$ points on a line where there are groups of arcs with the same left point (but no arcs with the same right point) and the group sizes would be given by \mathbb{B} .

This bijects with (\mathbf{r}^*) as noted above.

$\{1, 7, 8\}\{2, 5, 6\}\{3, 4\}$ $\{1, 2, 8\}\{3, 4, 5\}\{6, 7\}$ $\{1, 2, 3\}\{4, 7, 8\}\{5, 6\}$
 $\{1, 2, 3\}\{4, 5\}\{6, 7, 8\}$ $\{1, 2\}\{3, 4, 8\}\{5, 6, 7\}$

Fig. 18. Weighted Catalan examples for (\mathbf{o})

(n) Catalan: n nonintersecting chords joining $2n$ points on a circle.

Generalization (figure 19): Nonintersecting opening multi-chords of sizes given by \mathbb{B} joining $2n$ points on the circumference of a circle. We define a multi-chord to be a group of linked connections between points on the circumference on the circle (as opposed to just a single connection between two points, as in a chord). An opening multi-chord is one which only has multiple connections at one of its nodes, and that node comes before its connected nodes when traversing the circle clockwise.

Obviously, this notion requires a fixed starting point on the circle; we'll arbitrarily choose the node closest to the angle 0 from the center of the circle.

An alternate version of this is to use non-intersecting "chord segments", where each segment is a_i chords in a sequence, and the chords join points evenly spaced around the circle. Here the positions of the matching opening and closing parenthesis gives the points connected by the chords, and the weight of the opening chord gives the number of chords connected. See [8] for the p -Catalan case of this.

This has a trivial bijection to (\mathbf{r}^*) , using our notion of opening and closing parenthesis. Each opening multi-chord is a left parenthesis of a given weight, and each node it is connected to are its closing parenthesis. The non-crossing of the chords corresponds to the proper matching of the parenthesis in nested fashion. A similarly trivial bijection can be seen by arranging the points from (\mathbf{o}) around a circle with an arbitrary starting point, such as angle 0.

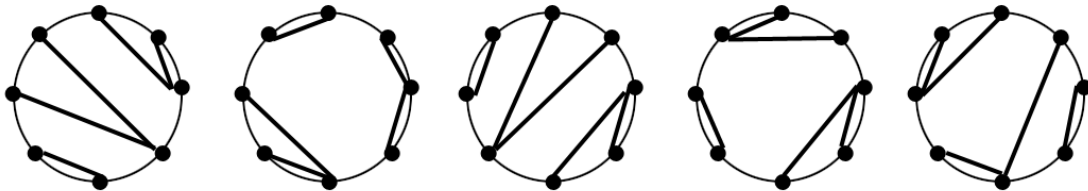


Fig. 19. Weighted Catalan examples for (\mathbf{n})

(\mathbf{p}) Catalan: Ways of drawing $n + 1$ points on a line with arcs connecting them such that the arcs do not pass below the line, the arcs are noncrossing, all the arcs at a given node exit in the same direction (left or right), and the graph thus formed is a tree.

Generalization (figure 20): As above, except that we add groupings to the arcs, such that the sizes of the groups are given by \mathbb{B} , and the arcs of a group are not separated

by an arc from a different group.

Bijection to **(r)**. Starting with the first node, count the number of left-to-right arcs coming out of it in each group, and write those numbers down from topmost group to bottommost. For every node after the first, start with a -1, then repeat the process of counting the groups. Don't count any zeros for nodes which only have right-to-left arcs. This gives a sequence of positive integers from \mathbb{B} and negative ones. Nonnegativity corresponds to the fact that we cannot have more nodes (after the first) than we have already counted left-to-right arcs.

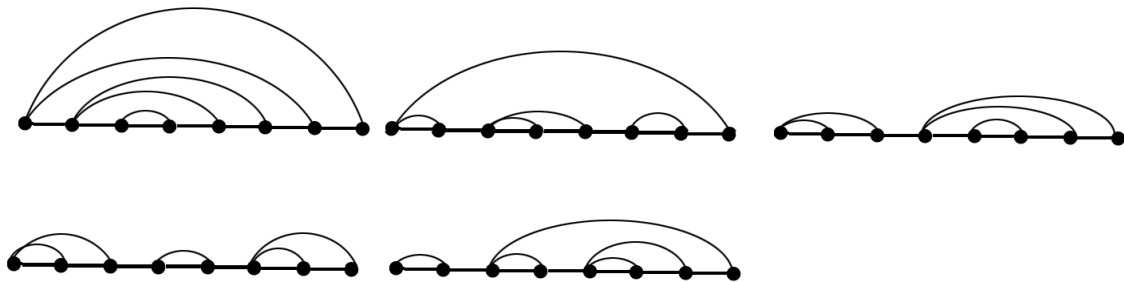


Fig. 20. Weighted Catalan examples for **(p)**

(a) Catalan: Partitions of an $(n + 2)$ -gon into triangles.

Generalization (figure 21): Partitions of an $(n + 2)$ -gon into polygons whose sizes are given by $a_i + 2$ where $a_i \in \mathbb{B}$.

Following the standard bijection for the Catalan numbers to **(d)**, we fix an edge as the base of the polygon, and fix a root of a tree in this edge. (In fig. 21 we have fixed the top edge as the base.) For a given partition of the $(n + 2)$ -gon, we place a node in each side of the partition containing the base, then connect each node to the root with a tree edge. This makes the root have degree equal to the number of sides in the polygon containing the base in our partition minus one. Then we treat each of the children as the root of their own subtree and continue recursively. The

degree of each node will correspond to an element of \mathbb{B} plus one, since after taking the base edge out of any polygon, it has that many remaining sides to form the children of its subtree's root node. Shapiro and Sulanke[21] showed this relationship over all partitions; this is the case specific to a chosen partition. (See fig.21 where we show the corresponding tree superimposed over the polygon partition below each polygon partition.)

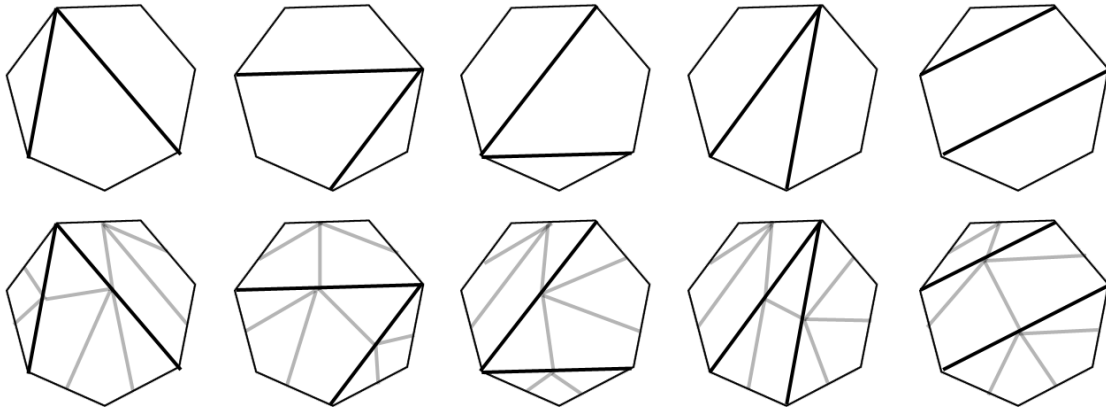


Fig. 21. Weighted Catalan examples for **(a)**

(pp) Catalan: Noncrossing partitions of $[n]$.

Generalization (figure 22): A pair of partitions of $[n]$, A and B , such that A is noncrossing, the blocks of B are of sizes given by \mathbb{B} , B is a subpartition of A , and inside any block of A , the elements of any block of B are continuous, *i.e.*, there do not appear in any block of A subsequences abc where a and c are members of the same block of B and b is not. In the Catalan case, our B is the trivial partition of singletons.

Bijection to **(r*)**: A block in the partition of A tells us how the opening parentheses are grouped for the close parentheses corresponding to the elements of the partition. For example, take the partition $12459 - 3 - 678$ for A with B being $124 - 3 - 5 - 6 - 78 - 9$. The first block in the A partition tells us that the first,

second, fourth, fifth, and ninth closing parenthesis have their opening parenthesis grouped together, and the B partition tells us that the first three have the same opening parenthesis, like so: $3((()*)*) **)$ (Where the stars represent locations of closing parentheses whose opening parenthesis have not yet been placed.) The 3 block in A gives us that the third parenthesis matches up to an open parenthesis of weight one, so the first $*$ can be replaced with $()$. The 678 block in the A partition and the corresponding subpartition in B tells us that we have the following construction near the end of the string: $(2))$) Thus, the full parenthesization is given by: $3((())(2))$)

$$\begin{aligned} \{1\}\{2,3\}\{4,5\} &\subset \{1,2,3,4,5\} & \{1,5\}\{2,3\}\{4\} &\subset \{1,5\}\{2,3\}\{4\} \\ \{12\}\{3\}\{4,5\} &\subset \{12\}\{3,4,5\} & \{1,2\}\{3\}\{4,5\} &\subset \{1,2\}\{3\}\{4,5\} \\ \{1\}\{2,5\}\{3,4\} &\subset \{1\}\{2,5\}\{3,4\} \end{aligned}$$

Fig. 22. Weighted Catalan examples for **(pp)**

(qq) Catalan: A partition A of $[n]$ such that if the numbers are arranged in order around a circle, then the convex hulls of the blocks of a given partition are pairwise disjoint.

Generalization: Pairs of partitions A and B of $[n]$ such that, if the numbers are arranged in order around a circle then the convex hulls of the blocks of a given partition are pairwise disjoint, B is a subpartition of A , and within any block of A , the subblocks of B consist of consecutive elements.

These are the same partitions as **(pp)**. The circle arrangement is another way to specify that the partitions are noncrossing.

(rr) Catalan: Noncrossing Murasaki diagrams with n vertical bars.

Generalization (figure 23): Noncrossing Murasaki diagrams with n vertical bars such that an additional grouping of the bars is accomplished by noncrossing underlines

which group the bars into discrete groups whose sizes are given by \mathbb{B} and the underlines are subsets of the overlines. In the Catalan case, each bar is just its own underline group.

This is another way to represent **(pp)**.



Fig. 23. Weighted Catalan examples for **(rr)**

(uu) Catalan: Nonnesting partitions of $[n]$.

Generalization (figure 24): A pair of partitions A and B of $[n]$ such that A is nonnesting, the block sizes of B are given by \mathbb{B} , B is a subpartition of A , and inside any block of A , the elements of any block of B are continuous, *i.e.*, there do not appear in any block of A subsequences abc where a and c are members of the same block of B and b is not.

This is in bijection to **(pp)**, using the natural bijection between non-crossing and non-nesting partitions.

$$\begin{aligned} \{1, 4\}\{2, 5\}\{3\} &\subset \{1, 4\}\{2, 5\}\{3\} & \{1, 2\}\{3, 4\}\{5\} &\subset \{1, 2\}\{3, 4\}\{5\} \\ \{1, 2\}\{3, 5\}\{4\} &\subset \{1, 2, 3, 5\}\{4\} & \{1, 2\}\{3\}\{4, 5\} &\subset \{1, 2, 3, 4, 5\} \\ \{1\}\{2, 3\}\{4, 5\} &\subset \{1, 2, 3\}\{4, 5\} \end{aligned}$$

Fig. 24. Weighted Catalan examples for **(uu)**

D. Permutations

(cc) Catalan: Permutations of the multiset $\{1^2, 2^2, \dots, n^2\}$ such that the first occurrences of each number appear in increasing order, and there is no subsequence of the form $abab$.

Generalization (figure 25): Permutations of the multisets $1^{b_1+1}, 2^{b_2+1}, \dots, k^{b_k+1}$ where the b_i are any arrangement of the elements of \mathbb{B} , the first occurrences of each number appear in increasing order, and there is no subsequence of the form $abab$.

This representation is in bijection to (\mathbf{r}^*) , with the first occurrence of each number being an open parenthesis, and the later occurrences being the closing parenthesis. The weight of the open parenthesis is given by the b_i in the multiset, and the subsequence restriction corresponds to the noncrossing condition on the parenthesis groups.

$$\begin{aligned} (1, 2, 3, 3, 2, 2, 1, 1) & \quad (1, 1, 2, 2, 2, 3, 3, 1) & \quad (1, 1, 1, 2, 3, 3, 2, 2) \\ (1, 1, 1, 2, 2, 3, 3, 3) & \quad (1, 1, 2, 2, 3, 3, 3, 2) \end{aligned}$$

Fig. 25. Weighted Catalan examples for **(cc)**

(dd) Catalan: Permutations $[2n]$ such that the odd numbers appear in increasing order, the even numbers appear in increasing order, and odd a appear before $a + 1$.

Generalization (figure 26): Permutations of the multisets $\{1, 2^{b_1}, 3, 4^{b_2}, \dots, (2m - 1), (2m)^{b_m}\}$ where the b_i are any arrangement of the elements of \mathbb{B} , with the restrictions above.

We biject this to (\mathbf{r}) by reading each odd number a as a positive integer (whose size is the b_i from the weight of $a + 1$ in the multiset), and each $a + 1$ as a -1 . The restrictions on order of appearance correspond to the partial sum restrictions of (\mathbf{r}) .

(ee) Catalan: 321-avoiding permutations of $[n]$.

(1, 3, 5, 2, 2, 4, 4, 6) (1, 2, 3, 2, 4, 5, 4, 6) (1, 2, 2, 3, 5, 4, 4, 6)
 (1, 2, 2, 3, 4, 5, 6, 6) (1, 2, 3, 4, 5, 4, 6, 6)

Fig. 26. Weighted Calatan examples for **(dd)**

Generalization: 321 avoiding permutations of $[n]$, such that the elements of the permutation are partitioned into contiguous blocks whose sizes are given by \mathbb{B} .

Bijection: These are the same permutations as **(jj)** (see figure on page 60); the restrictions on the blocks and the 321 avoidance matches the restrictions on them in **(jj)** without referencing the queues.

(gg) Catalan: Permutations w of $[2n]$ with n cycles of length two such that the product $(1, 2, \dots, 2n)w$ has n cycles.

Generalization (figure 27): Permutations w of $[m + n]$ with cycle lengths given by \mathbb{B} such that the product $(1, 2, \dots, 2n)w$ has m cycles.

Bijection with **(r*)**: Take a valid parenthesis arrangement, and label each parenthesis from 1 to k in order of appearance. Then for each open parenthesis, write down the parentheses in that group in decreasing order of appearance as a cycle. For example, if a parentheses group has its open parenthesis at position 4, and its closing parentheses at positions 5, 8, and 12, then its corresponding cycle will be $(12, 8, 5, 4)$. This gives a permutation of cycles with appropriate size according to \mathbb{B} .

Now, if a parenthesis c_j of one group precedes the first parenthesis of a different group, d_1 , then the cycle in the product containing c_k will have the sequence c_k, d_ω where d_ω is the final parenthesis of the group that d_1 starts.

Furthermore, if we have the last parenthesis c_ω of a group followed by some parenthesis d_i of a different group, then the cycle in the containing c_ω will have the sequence c_ω, d_{i-1} , where d_{i-1} will be d_ω if d_i is the first parenthesis of its group (aka, d_1).

Finally, any parenthesis c_j followed by c_{j+1} from its own group will form a singleton in the product (as we will have c_j, c_{j+1} in the first part of the product).

Using these facts, we can label groups of parentheses by level, so that the groups which are completely contiguous are on level 0, groups that only contain groups of level 0 are on level 1, groups that only contain groups of level less than i are on level i .

Then we see that a level 0 group with opening parenthesis of weight f will become f singletons in the final product (every element but the last will become a singleton), and the final element of the level 1 group will be linked into the preceding and following groups.

For a level 1 group of weight g , we see that the elements preceding the level 0 groups will be linked in a cycle with the final element of the level 0 groups, and the final element will be linked into the preceding and following groups, and the other elements will become single cycles. If such a group contains m level 0 groups of weights $c_1 \dots c_m$, then the entirety will contain $\sum_{i=1}^m c_i$ singletons from the level 0 groups, $g - m$ singletons from the level 1 group and m groups that are either links into the level 0 groups or singletons (depending on how many of the level 0 groups follow a different level 0 group instead of an element of our level 1 group). The final element of this group will link into the preceding and following groups, but the rest of the elements give us a number of cycles in the final product exactly equal to the weights of the opening parenthesis contained within.

We can continue by induction on groups of higher level, seeing that each time, we will get a number of cycles in the final product equal to the sums of the weights of the opening parenthesis of the contained groups, with one element linking to outside the group.

When we reach the highest level, that final link out completes a cycle, giving us

the required $m + 1$ cycles in the product.

Details of the inverse map are omitted other than to note that the requirement for the number of cycles in the product enforces the non-crossing nature of the partition and the ordering of the pre-product cycles, and the size of the pre-product cycles gives us groups of the appropriate weight.

$$\begin{array}{lll} (1, 8, 7)(2, 6, 5)(3, 4) & (1, 8, 2)(3, 5, 4)(6, 7) & (1, 3, 2)(4, 8, 7)(5, 6) \\ (1, 3, 2)(4, 5)(6, 8, 7) & (1, 2)(3, 8, 4)(5, 7, 6) & \end{array}$$

Fig. 27. Weighted Catalan examples for **(gg)**

(ii) Catalan: Permutations of $[n]$ that can be stack sorted.

Generalization (figure 28): Permutations of $[n]$ with the elements of the permutation partitioned into contiguous blocks whose sizes are given by \mathbb{B} , such that the partition is stack sortable if the elements in a block must be moved onto the stack together. (They can be removed individually.)

Bijection to **(r)** by recording the sizes of the blocks moved onto the stack as positive numbers and removals from the stack as -1 .

$$(\underline{5}, \underline{4}, \underline{3}, \underline{2}, 1) \quad (\underline{5}, \underline{1}, \underline{3}, \underline{2}, 4) \quad (\underline{2}, \underline{1}, \underline{5}, \underline{4}, 3) \quad (\underline{2}, \underline{1}, 3, \underline{5}, \underline{4}) \quad (1, \underline{5}, \underline{2}, \underline{4}, \underline{3})$$

Fig. 28. Weighted Catalan examples for **(ii)**

(jj) Catalan: Permutations of $[n]$ that can be sorted on two parallel queues. (See Stanley[25])

Generalization (figure 29): Permutations of n such that elements of the permutation are partitioned into blocks whose size are given by \mathbb{B} , that can be sorted on two parallel queues if the elements in a block must be moved onto a queue together without any intervening removals from the queues being allowed.

Bijection to **(r)**: Whenever a group of elements is moved onto the queues, record a positive integer of the size of the group. Whenever an element is removed from a queue, record a -1 .

$(\underline{2}, \underline{3}, \underline{4}, \underline{5}, 1)$ $(\underline{1}, \underline{5}, \underline{2}, \underline{3}, 4)$ $(\underline{1}, \underline{2}, \underline{4}, \underline{5}, 3)$ $(\underline{1}, \underline{2}, 3, \underline{4}, \underline{5})$ $(1, \underline{2}, \underline{5}, \underline{3}, \underline{4})$

Fig. 29. Weighted Catalan examples for **(jj)**

(kk) Catalan: Permutations of $[2n]$ consisting of cycles of length 2 such that the cycles form a noncrossing partition.

Generalization: Permutations of $[m+n]$ whose cycle lengths are given by \mathbb{B} , such that the cycles form a non-crossing partition of $[m+n]$ and each cycle is in decreasing order.

At the same time, these are both the cycles that will satisfy **(gg)** and the partitions that will satisfy **(o)** (with a specific ordering given for listing each block). (See figure 27.)

(aaa) Catalan: Linear extensions of the poset $[2] \times [n]$

Generalization (figure 30): Create a poset by taking the elements of the multiset $\{1, 2^{b_1}, 3, 4^{b_2}, \dots\}$ where the b_i are any arrangement of the elements of \mathbb{B} (as in **(dd)**) with the partial ordering $<_!$ such that if a and b are both odd or both even then $a < b \rightarrow a <_! b$. If $a = c_i$ and $b = c_j$ then $i < j \rightarrow a <_! b$. And if a is odd and $b = a + 1$ then $a <_! b$. The elements counted by our formula are the linear extensions of these posets.

Bijection: The posets correspond to the multisets of **(dd)** and the linear extensions of the posets are the restricted permutations of **(dd)**. One can also link directly into **(r)** by reading the odd numbers as positive integers (whose weight is given by their corresponding even number) and the even numbers as -1 's.

1, 3, 5, 2₁, 2₂, 4₁, 4₂, 6 1, 2₁, 3, 2₂, 4₁, 5, 4₂, 6 1, 2₁, 2₂, 3, 5, 4₁, 4₂, 6
 1, 2₁, 2₂, 3, 4, 5, 6₁, 6₂ 1, 2, 3, 4₁, 5, 4₂, 6₁, 6₂

Fig. 30. Weighted Catalan examples for **(aaa)**

E. Other Equivalent Representations

(b) Catalan: Binary parenthesizations of a string of $n + 1$ letters.

Generalization (figure 31): Parenthesizations of $n + 1$ letters into blocks given by \mathbb{B} ; *i.e.*, each pair of parenthesis encloses b_i (where $b_i \in \mathbb{B}$) items where each item is either a letter or another parenthesization.

Bijection to **(d)**: Given such a parenthesization, we look at the outermost group of items, and the size of this group gives us the degree of the root. Then we look at the rightmost item in this grouping, and we look at each item as a group, with a letter designating a leaf, and a sub-parenthesization representing a subtree. For example, the parenthesization $((xx)xx)$ has an outermost group of size 3, giving a root of degree three. Its first child is a subtree whose root is of degree two with two leaves, and its second and third child are leaves. Contrast this with $(x(xxx))$ whose outermost group designates a root of degree two whose first child is a leaf and whose second child is a node with three children, all leaves.

$((xx)xx)xx$ $x(xx(xx))x$ $xx((xx)xx)$ $xx(x(xxx))$ $x(x(xxx)x)$

Fig. 31. Weighted Catalan examples for **(b)**

(s) Catalan: Sequences $1 \leq a_1 \leq \dots \leq a_n$ with $a_i < i$.

Generalization (figure 32): Sequences $\langle a_i \rangle$ of n numbers such that they are weakly increasing, grouped into contiguous groups of identical numbers of sizes given by \mathbb{B} , and each $a_i \leq i$.

Bijection with (\mathbf{h}) : These are the heights plus one of (\mathbf{h}) 's $x = i - 1$, and the groupings give the size of the horizontal steps.

11 11 1 11 22 3 11 22 2 11 2 33 1 22 33

Fig. 32. Weighted Calatan examples for (\mathbf{s})

(\mathbf{ww}) Catalan: Standard Young Tableaux of shape (n, n) .

Generalization (figure 33): The elements of $n + m$ laid out in a diagram whose first row is of width m , whose column heights are given by the weights in \mathbb{B} , whose rows and columns are in increasing order, and elements not in the first row are ordered among columns such that each is greater than the elements appearing in the columns to its left. Note that this differs from a Young Tableaux in that there may be gaps between elements in the later rows, and we have a stronger restriction on ordering.

Bijection to (\mathbf{r}) : The first row gives the location of the positive numbers (of size equal one less than the column's height) and the other elements give the location of the -1 's. The increasing column restriction is equivalent to the nonnegativity restriction on (\mathbf{r}) and the fill requirement for the lower rows ensures a unique representation for each sequence.

123	136	145	146	135
468	258	268	257	247
57	47	37	3 8	68

Fig. 33. Weighted Calatan examples for (\mathbf{ww})

F. Two-Sided Generalization

In the representations of our generalized Catalan numbers given above, we allowed one side to vary the weights of its elements, while requiring the other side to remain

fixed. However, one can consider a two-sided ballot problem, where both positive and negative numbers can be of any weight, so long as the sum total is equal to zero, and the valid arrangements of a weight set require nonnegative partial sums. We represent this with a new weight set \mathbb{C} as a multiset over \mathbb{Z} with $0 \notin \mathbb{C}$ and the sums of the positive and negative weights are the same. This generalization has been investigated by Curtis Coker[3], who shows that the generating function $\sum_{k=1}^n \frac{1}{n} \binom{n}{k} \binom{n}{k-1} x^{2k} (1+x)^{2n-2k}$ gives us the number of structures with the coefficients for x^i giving the total ballot arrangements whose partitions have i blocks. Plugging in $x = 1$ in the above formula gives us $\sum_{k=1}^n 4^{n-k} N(n, n-k)$ where $N(n, i)$ are the Narayana numbers. This is the number of all two-sided structures whose total length is n (See Sloane's A059231[22]).

CHAPTER V

SUMMARY

We have discovered a formula for the descents in parking functions and a generating function for them based on the number of ties. We have also discovered a generating function for the number of inversions in parking functions with a given weight set. In the process, we have found an extension to the Catalan numbers which allows for different weights to be combined into one structure. Finally, we have explored extensions to parking functions, gave a recursion for the linear probes in blocked parking functions, and gave generating functions for the first few cases.

REFERENCES

1. J. S. Beissinger and U. N. Peled, *A note on major sequences and external activity in trees*, The Electronic Journal of Combinatorics **3** (1996).
2. P. Biane, *Parking functions of types a and b* , The Electronic Journal of Combinatorics **9** (2002).
3. C. Coker, *Enumerating a class of lattice paths*, Discrete Math **271** (2003), 13–28.
4. A.D. Foata and J. Riordan, *Mappings of acyclic and parking functions*, Aequationes Mathematicae **10** (1974), 10–22.
5. J. D. Gilbey and L. H. Kalikow, *Parking functions, valet functions, and priority queues*, Discrete Mathematics **197-198** (1999), 351–373.
6. J. Haglund and N.A. Loehr, *A conjectured combinatorial formula for the Hilbert series for diagonal harmonics*, Discrete Mathematics **298** (2005), 189–204.
7. M. Haiman, *Conjectures on the quotient ring by diagonal invariants*, Journal of Algebraic Combinatorics **3** (1994), 17–76.
8. S. Heubach, N. Li, and T. Mansour, *A garden of k -Catalan structures*, <http://www.calstatela.edu/faculty/sheubac/papers/k-Catalan%20structures.pdf> (Accessed on May 1 2009).
9. P. Hilton and J. Pedersen, *Catalan numbers, their generalization, and their uses*, Math. Intelligencer **13** (1991), 64–75.
10. D. A. Klarner, *Correspondences between plane trees and binary sequences*, Journal of Combinatoric Theory **9** (1970), 401–411.

11. D. Knuth, *The Art of Computer Programming*, Addison-Wesley, Reading, Massachusetts, 1997.
12. ———, *Linear probing and graphs*, *Algorithmica* **22** (1998), 561–568.
13. A. G. Konheim and B. Weiss, *An occupancy discipline and applications*, *SIAM Journal on Applied Mathematics* **14** (1966), 1266–1274.
14. G. Kreweras, *Une famille de polynomes ayant plusieurs propri tes enumeratives*, *Periodica Mathematica Hungaria* **11** (1980), 309–320.
15. J. Kung, X. Sun, and C. H. Yan, *Goncarov-type polynomials and applications in combinatorics*, Preprint (2005).
16. J. Kung and C. H. Yan, *Exact formula for moments of sums of classical parking functions*, *Advances in Applied Mathematics* **31** (2003), 215–241.
17. ———, *Expected sums of moments general parking functions*, *Annals of Combinatorics* **7** (2003), 481–493.
18. ———, *Goncarov polynomials and parking functions*, *Journal of Combinatorial Theory* **102** (2003), no. 1, 16–37.
19. N.A. Loehr, *Combinatorics of q,t -parking functions*, *Advances in Applied Math* **34** (2005), 408–425.
20. P. Peart and W. Woan, *Dyck paths with no peaks at height k* , *Journal of Integer Sequences* **4** (2001).
21. L. Shapiro and R. Sulanke, *Bijections for the Schr oder numbers*, *Mathematics Magazine* **73** (2000), 369–376.

22. N.J.A. Sloane, *Online encyclopedia of integer sequences*, <http://www.research.att.com/~njas/sequences/>.
23. R. P. Stanley, *Parking functions and noncrossing partitions*, *The Electronic Journal of Combinatorics* **4** (1997).
24. ———, *Hyperplane arrangements, parking functions, and tree inversions*, *Mathematical Essays in Honor of Gian-Carlo Rota* **161** (1998), 359–375.
25. ———, *Enumerative Combinatorics*, vol. 2, Cambridge University Press, New York, New York, 2005.
26. ———, *Enumerative Combinatorics*, vol. 1, Cambridge University Press, New York, New York, 2005.
27. ———, *Exercises on Catalan and related numbers*, <http://math.mit.edu/~rstan/ec/catalan.pdf> (accessed on May 1 2009), 2009.
28. C. H. Yan, *Generalized tree inversions and k -parking functions*, *Journal of Combinatorial Theory* **79** (1997), 268–280.
29. ———, *On the enumeration of generalized parking functions*, *Congressus Numerantium* (2000), no. 147, 201–209.
30. ———, *Generalized parking functions, tree inversions and multicolored graphs*, *Advances in Applied Mathematics* **27** (2001), 641–670.
31. C. H. Yan and D. Kostic, *Multiparking functions, graph search, and tutte polynomials*, *Advances in Applied Mathematics* **40** (2008), 73–97.

VITA

Paul R. F. Schumacher was born in 1977. He attended the University of Chicago, and graduated in 1999 with bachelor's degrees in mathematics, computer science, and statistics. He can be reached via Texas A&M Department of Mathematics, Texas A&M University, College Station, TX 77843-3368. His email address is schumach@math.tamu.edu.