ENUMERATING k-NAPLES PARKING FUNCTIONS THROUGH CATALAN OBJECTS

João Pedro Carvalho
Department of Mathematics, University of Michigan, Ann Arbor, Michigan
jpcarv@umich.edu

Pamela E. Harris
Dept. of Mathematical Sciences, University of Wisconsin, Milwaukee, Wisconsin
peharris@uwm.edu

Gordon Rojas Kirby
Department of Mathematics, San Diego State University, San Diego, California
gkirby@sdsu.edu

Nico Tripeny
Haverford College, Haverford, Pennsylvania
ntripeny@haverford.edu

Andrés R. Vindas-Meléndez
Department of Mathematics, University of California, Berkeley, California
and
Department of Mathematics, Harvey Mudd College, Claremont, California
andres.vindas@berkeley.edu; arvm@hmc.edu

Received: 9/10/21, Revised: 11/28/23, Accepted: 4/23/24, Published: 5/20/24

Abstract

This paper studies a generalization of parking functions named k-Naples parking functions, where backward movement is allowed. One consequence of backward movement is that the number of ascending k-Naples is not the same as the number of descending k-Naples. This paper focuses on generalizing the bijections of ascending parking functions with combinatorial objects enumerated by the Catalan numbers in the setting of both ascending and descending k-Naples parking functions. These combinatorial objects include Dyck paths, binary trees, triangulations of polygons, and non-crossing partitions. Using these bijections, we enumerate both ascending and descending k-Naples parking functions.

1. Introduction

Parking functions are special types of integer sequences that were proposed independently by Ronald Pyke [7] as well as by Alan Konheim and Benjamin Weiss

DOI: 10.5281/zenodo.11221617

[6] in order to study hashing problems in computer science. If we have a sequence of n integers all belonging to $[n] := \{1, 2, \ldots, n\}$, we call it a *parking preference* of length n. A *parking function* of length n is a special type of parking preference $(a_1, a_2, \ldots, a_n)$, that allows n cars $c_1, c_2, \ldots, c_n$ with respective preferences $a_1, a_2, \ldots, a_n$ to park in a one-way street with n consecutively ordered parking spots according to the following rules:

1. c_1 parks in its preferred spot;
2. every new car parks in its preferred spot if it is not occupied, otherwise it parks in the next available spot.

For example, the parking preference $(2, 2, 1, 4)$ is a parking function of length 4, where c_1 parks in the second spot, c_2 in the third, c_3 in the first, and c_4 in the fourth.

Special subsets of parking functions are the monotonic ones, that correspond to *ascending* (weakly increasing) or *descending* (weakly decreasing) parking preferences. The set of ascending parking functions of length n, as well as the set of descending parking functions of length n, are counted by the Catalan numbers. There are many well-known bijections between either of these subsets of parking functions and a variety of Catalan objects, including Dyck paths, binary trees, triangulations of n-gons, and noncrossing partitions [8]. We remark that the equinumerosity of ascending and descending parking functions follows from the fact that if a given parking preference is a parking preference, then so are all of its rearrangements.

Many generalizations to parking functions were proposed throughout the years, including allowing cars with different lengths and starting with some spots already filled, and a reader interested in exploring different directions may find [2] a useful resource. In this paper, we focus our attention on the generalization known as *Naples parking functions*. First proposed in [1], Naples parking functions differ from standard parking functions by the rules in which the cars park. In the Naples parking scheme, cars $c_1, c_2, \ldots, c_n$ park one at a time by first checking their preferred spot. If the preferred spot was already occupied, then cars check one spot before, parking there if available, otherwise cars move forward to the next available spot if that space is occupied. A parking preference is a Naples parking function if all cars park successfully according to this Naples parking scheme. Similarly, the *k-Naples parking scheme* allows cars to first check their preferred spot, then check up to k spots preceding their parking preference, in decreasing order, before moving forward to the next available space. A parking preference is a *k-Naples parking function* if all cars park successfully according to this k-Naples parking scheme. For example, the parking preference $(6, 6, 6, 5, 5, 2, 1)$ is a 2-Naples parking function as the cars $c_1, \ldots, c_7$ park in positions $6, 5, 4, 3, 7, 2, 1$, respectively, with the second car moving back once and the third and fourth cars moving back twice since both their

preferred spot and the one directly behind it are taken. Moreover, $(6, 6, 6, 5, 5, 2, 1)$ is a k-Naples parking function for $k \geq 3$, where cars $c_1, \ldots, c_7$ park in positions $6, 5, 4, 3, 2, 1, 7$. We see that this is not a 1-Naples parking function as car c_5 cannot park in that setting.

In fact, we have the following proposition, proved in Section 2.

Proposition 1. *If α is a k-Naples parking function of length n, then it is also a $(k+1)$-Naples parking function of length n.*

Remark 1. As a consequence of Proposition 1, the notion of k-Naples for $0 \leq k \leq n-1$ offers an interpolation between ordinary parking functions when $k = 0$ and all n-tuples of integers belonging to $[n]$ when $k \geq n-1$. We remark that we refer to 1-Naples parking functions simply as Naples parking functions. However, unless otherwise specified, we adopt the convention that $k \geq 1$ for all k-Naples functions considered in this paper.

We now summarize our main results in this paper. As we commented previously, all rearrangements of a parking function are parking functions. In Section 2 we answer the analogous question for k-Naples parking functions, by establishing that given a parking preference, all of its rearrangements are k-Naples parking functions if and only if its ascending rearrangement is a k-Naples parking function. This is the statement of Theorem 1. We then restrict our study to ascending and descending k-Naples parking functions, and in Section 3 we present bijections between ascending and descending k-Naples parking functions and families of Dyck paths. In Section 4 we use the bijections found in the previous sections to give formulas to enumerate ascending and descending k-Naples parking functions. These results give connections to Fine numbers (OIES sequence A000957) and convolution of the Catalan numbers with the Fine numbers (OIES sequence A000958). We also state other bijections between ascending and descending k-Naples parking functions and other Catalan objects. This is the content of Section 5. Following these results, we conclude the article in Section 6 by detailing some future directions of research on these topics.

2. k-Naples Parking Functions and Their Rearrangements

Before proving Proposition 1 we note that by adapting the technique from the recursive formula for k-Naples parking functions from [3] we can inductively show that a k-Naples parking function of length n is also a $(k+1)$-Naples parking function of the same length.

First, we prove the following lemma. It states that if a k-Naples parking function of length n has the last car c_n park in space $m+1$, then the cars that parked after

spot $m+1$ will all park somewhere in spots $m+1,\ldots,n$ according to the $(k+1)$-Naples parking scheme. In order to prove this lemma, consider the following setup. Let the parking lot be represented by a number line of integers so that cars can reverse past spot 1 or continue forward past spot n. In this way, we say that a car parks on $\mathbb{Z}$. For example, for parking preference $\alpha = (4,2,2,2)$, parking according to the 2-Naples parking scheme, the cars park on $\mathbb{Z}$ as illustrated in Figure 1.

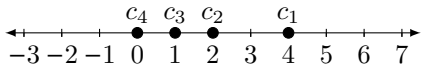

Figure 1: Parking position of cars with parking preference $\alpha = (4,2,2,2)$.

Lemma 1. *Suppose that $\alpha \in [n]^n$ is a parking preference for which, according to the k-Naples parking scheme, the n cars $c_1,\ldots,c_n$ park on $\mathbb{Z}$ in spots $1,\ldots,n$. Then each c_i parks on $\mathbb{Z}$ according to the $(k+1)$-Naples parking scheme in a spot $0 \le s \le n$.*

Proof. We proceed by induction on n. If $n = 1$, then $\alpha = (1)$. For any $k \ge 0$, according to the k-Naples parking scheme, car c_1 parks on $\mathbb{Z}$ in spot 1. Under the $(k+1)$-Naples parking scheme, car c_1 parks on $\mathbb{Z}$ in spot $s = 1$, which satisfies $0 \le s \le 1$. Suppose the result holds true for all parking preferences of length less than n. Let $\alpha = (a_1,\ldots,a_n) \in [n]^n$ be a parking preference for which n cars $c_1,\ldots,c_n$ park, respectively, on $\mathbb{Z}$ according to the k-Naples parking scheme and occupy spots $s_1,\ldots,s_n \in [n]$, with $s_n = m+1$ and $0 \le m \le n-1$. Then $a_n \le m+k+1$. Define S to be the set of cars parking before spot $m+1$ and T to be the set of cars parking after spot $m+1$. Then $\min(k+1, m+1) \le a_n$, $a_i \le m$ for all i such that $c_i \in S$, and $a_j \ge m+2$ for all j such that $c_j \in T$.

Observe that cars $c_i \in S$ never check a spot after m, and cars $c_j \in T$ never check a spot before $m+2$ so that we may consider decomposing α into two disjoint ordered lists of preferences that do not interact and satisfy our induction hypothesis. That is, if we allow cars $c_1,\ldots c_n$ with parking preference α to park on $\mathbb{Z}$ according to the $(k+1)$-Naples parking scheme in spots s_i', respectively. Then $\{s_i' \mid c_i \in S\} \subset \{0,\ldots,m\}$ and $\{s_j' \mid c_j \in T\} \subset \{m+1,\ldots,n\}$. Hence, there is some parking spot between 0 and n that is unoccupied on the lot when c_n goes to park according to the $(k+1)$-Naples parking scheme.

If $a_n < k+1$, then $a_n \ge m+1$ and either there is a spot in $[m]$ available in which c_n parks or c_n parks in spot 0 according to the $(k+1)$-Naples parking scheme. If $a_n \ge k+1$ then there is some spot $m+1,\ldots,n$ available for c_n to park in. Hence, c_n parks in some spot between 0 and n according to the $(k+1)$-Naples parking scheme. □

Remark 2. Note that in the last sentence of the above proof, when $a_n \ge k+1$ there

may additionally be a spot $m-k-1,\ldots,m$ open that c_n parks in according to the $(k+1)$-Naples parking scheme before proceeding to the available spot between space $m+1$ and space n. For example, consider the parking preference $(1,2,2,5,7,7,7,3)$ of length 8 parking according to the 2-Naples parking scheme on $\mathbb{Z}$. The cars park in spots $1,2,\ldots 8$, and the last car parks in spot 4. However, in the 3-Naples scheme on $\mathbb{Z}$, cars $c_1,\ldots,c_8$ park in spots 1,2,0,5,7,6,4, and 3, respectively. Specifically, when c_8 goes to park it has spot 8 and spot 3 available. It checks spot 3 first so it parks there.

Proof of Proposition 1. We proceed by induction on n. If $n = 1$, then $\alpha = (1)$, which is a k-Naples parking function for all $k \geq 0$, in which car c_1 parks on $\mathbb{Z}$ in spot 1. Under the $(k+1)$-Naples parking scheme car c_1 parks on $\mathbb{Z}$ in spot $s = 1$. Hence α is a $(k+1)$-Naples parking function. Suppose that every k-Naples parking function of length $1 \leq l < n$ is a $(k+1)$-Naples parking function of length l. Let $\alpha = (a_1,\ldots,a_n)$ be a k-Naples parking function of length n with cars $c_1,\ldots,c_n$ parking in spots $s_1,s_2,\ldots,s_n$, respectively, with $s_n = m+1$ and $0 \leq m \leq n-1$. Then $a_n \leq m+k+1$. Define S to be the set of cars parking before spot $m+1$ and T to be the set of cars parking after spot $m+1$. Then $a_i \leq m$ for all i such that $c_i \in S$, and $a_j \geq m+2$ for all j such that $c_j \in T$.

Observe that cars $c_i \in S$ never check a spot after m, and cars $c_j \in T$ never check a spot before $m+2$, so that we may consider decomposing α into disjoint ordered lists of preferences. The preferences of cars that parked after spot $m+1$ according to the k-Naples parking scheme satisfy Lemma 1, so that they only check and park in spots $m+1$ or after according to the $(k+1)$-Naples parking scheme. Hence, we may consider the preferences of cars in S, those that parked before spot $m+1$ according to the k-Naples scheme, to be a k-Naples parking function of length $m < n$, satisfying our induction hypothesis so that they park in spots $1,\ldots,m$ according to the $(k+1)$-Naples parking scheme.

Hence, there is an unoccupied spot between $m+1$ and n when c_n goes to park. Since $a_n \leq m+k$, it is able to successfully park in the unoccupied space according to the $(k+1)$-Naples parking scheme. Therefore, α is a $(k+1)$-Naples parking function. □

Another interesting question regarding k-Naples parking functions is if all the rearrangements of k-Naples parking functions remain k-Naples parking functions, since this is true for traditional parking functions. However, this is not the case, as we illustrate in Example 1. The bijective correspondence between ascending and descending parking functions does not hold in the k-Naples case for $k > 0$. This motivates us to study both ascending and descending k-Naples parking functions separately to understand their similarities and differences in the hopes of achieving a better understanding of general k-Naples parking functions.

Example 1. We have $(6,6,6,5,5,2,1)$ as a descending parking preference of length 7 and $(1,2,5,5,6,6,6)$ as its corresponding ascending parking preference. From above, we see that $(6,6,6,5,5,2,1)$ is a descending 2-Naples parking function. We also can see that $(1,2,5,5,6,6,6)$ is an ascending 3-Naples parking function but not an ascending 2-Naples parking function.

As a starting point to begin exploring exactly when rearrangements of a k-Naples parking function are still k-Naples, we define a slightly modified way of writing parking preferences that can capture the position of cars step by step as they park. Then, we prove a lemma that expose small modifications in the parking preference that conserve its status as a parking function.

Definition 1. An *i-filled* parking preference of length n is an ordered pair of sequences in $[n]$, of the form $((d_1,\ldots,d_i),(a_{i+1},\ldots,a_n))$, such that each d_j for $1 \le j \le i$ represents the spot in which car c_j has already parked, and each a_j for $i < j \le n$ represents the preference of car c_j that has yet to park. If all cars can park using the k-Naples rules we call this an *i-filled* k-Naples parking function.

Lemma 2. *If $P_1 = (a_1, a_2, \ldots, a_n)$ is a k-Naples parking function with cars parking in spots $(d_1,\ldots,d_n)$ and we consider the i-filled parking preference*

$$P_2 = ((p_1,\ldots,p_i),(a_{i+1},\ldots,a_n))$$

with $p_j = d_j$ for all but exactly one $l \le i$ where $p_l < d_l$, then P_2 is an i-filled k-Naples parking function.

Proof. We prove this by induction on $n-i$. Let $n-i=1$ or $i=n-1$. Suppose that P_2 is an $(n-1)$-filled parking preference $((p_1,\ldots,p_{n-1}),(a_n))$ so that, according to this preference, cars $c_1,\ldots,c_{n-1}$ park in spots $p_1,\ldots p_{n-1}$. By assumption, $p_j = d_j$ for all $j \le i$ except for one $1 \le l \le n-1$ where $p_l < d_l$. Since all cars must park in distinct spots we must have $d_n = p_l < d_l$ so that spot d_l is unoccupied when c_n goes to park and c_n is able to park in spot d_l. Thus, P_2 is an $(n-1)$-filled parking preference.

Now, suppose the same is true for every i such that $k < i < n$ and suppose that P_2 is a k-filled parking preference $((p_1,\ldots,p_k),(a_{k+1},\ldots a_n))$ so that, according to this preference, cars $c_1,\ldots,c_k$ park in spots $p_1,\ldots p_k$. By assumption, $p_j = d_j$ for all but exactly one $l \le i$, where $p_l < d_l$. Thus, $p_l = d_j$ for some $j > k$. Now consider where car c_{k+1} parks according to P_2.

If $p_l = d_{k+1}$, then d_l is unoccupied when c_{k+1} tries to park. By assumption, $d_{k+1} < d_l$, which forces c_{k+1} to park at spot d_l or earlier. If c_{k+1} parks in spot d_l, then when cars $c_{k+2},\ldots,c_n$ go to park according to P_2 they find spots $d_1,\ldots,d_{k+1}$ occupied and park in spots $d_{k+2},\ldots d_n$ respectively. Thus, we may assume that c_{k+1} parks between spots d_{k+1} and d_l. Hence, the collection of spots X occupied by cars $c_1,...,c_{k+1}$ differs as a set from $X' = \{d_1,\ldots,d_{k+1}\}$ by one element. Say

$X \setminus X' = d'$ and note $X' \setminus X = d_l$ so that $d' < d_l$. Then we can arrange these spots into a $(k+1)$-filled parking preference satisfying our inductive hypothesis so that it is a $(k+1)$-filled k-Naples parking function. But that means that cars $c_{k+2}, \dots c_n$ are able to park based on how cars $c_1, ..., c_{k+1}$ have filled the lot according to P_2 so that P_2 is a k-filled Naples parking function.

If $p_l \neq d_{k+1}$ then it must be the case that $p_l = d_j$ for some $j > k+1$. Then when c_{k+1} goes to park it can either park in d_l, d_{k+1}, or some earlier spot. If it parks in d_l we fall into the same situation as above of having a $(k+1)$-filled k-Naples parking function. Similarly, if c_{k+1} parks in d_{k+1}, then we have a $(k+1)$-filled k-Naples parking function satisfying the induction hypothesis. Lastly, if it parks in some earlier spot, this spot would have also been available to c_{k+1} when it tried to park according to P_1 and thus is a contradiction. □

Lemma 2 plays a key role in the following results about rearrangements of k-Naples parking functions.

Theorem 1. *Given a parking preference, all of its rearrangements are k-Naples parking functions if and only if its ascending rearrangement is a k-Naples parking function.*

Proof. Note that if all rearrangements of a parking preference are k-Naples then this includes the fact that the ascending rearrangement is k-Naples. To prove that all rearrangements of an ascending k-Naples parking function are k-Naples it suffices to show that if we have a k-Naples parking function $P_1 = (a_1, a_2, \dots, a_n)$, with $a_i < a_{i+1}$ for some i, then the preference $P_2 = (b_1, b_2, \dots, b_n)$ where $b_j = a_j$ for every $j \notin \{i, i+1\}$, $b_i = a_{i+1}$, and $b_{i+1} = a_i$, is also a k-Naples parking function.

Let car c_j park in spot d_j in accordance with parking preference P_1. We see that both parking preferences P_1 and P_2 result in the first $i-1$ cars park identically, then $p_j = d_j$. If c_i in P_2 parks in d_{i+1}, then c_i in P_1 must not pass by d_{i+1} before parking or it would park there. The two cars take up the same spaces in P_2 and the rest of the parking proceeds as in P_1. So, we may assume c_i in P_2 does not park in d_{i+1}. But then it must be parking in a spot not open to c_{i+1} in P_1, namely d_i. Then when c_{i+1} goes to park in P_2, it drives past d_i which is now full.

If $d_{i+1} < d_i$, we see that car c_{i+1} in P_1 backed up all the way to d_{i+1} and since $b_{i+1} < b_i$, c_{i+1} in P_2 backs up to d_{i+1} as well. Otherwise, we have $d_{i+1} > d_i$ and c_{i+1} in P_2 clearly parks at or before d_{i+1}. So, we know that after the $i+1$st car in P_2 parks, all the spots the first i cars park in for P_1 are full, and a car is either parked in d_{i+1} or in a spot that would be open in P_1 that is before d_{i+1}. Since this is the situation in the previous lemma, we see that P_2 is a k-Naples parking function as desired. □

Example 2. Observe that $(6, 6, 5, 5, 3, 1)$ is a 2-Naples parking function, but its rearrangement $(3, 5, 1, 6, 6, 5)$ is not. Note that in the ascending rearrangement

$(1, 3, 5, 5, 6, 6)$, no car can park in the second spot according to the 2-Naples parking scheme. So it is not a 2-Naples parking function. However, we know $(1, 3, 3, 5, 6, 6)$ is an ascending 2-Naples parking function, and all of its rearrangements are also 2-Naples parking functions.

Remark 3. The proof of Theorem 1 defines a hierarchy that is followed when deciding when rearrangements of a parking preference are k-Naples. If two rearrangements differ by just one switch, where the switched car that comes after in the first rearrangement has a higher preference than the one coming before, then it is intrinsically harder for the first one to be k-Naples than the second one. This is because according to the proof, the first being k-Naples implies the second also is, but the converse is not true.

3. Dyck Paths

One family of Catalan objects that is in bijection with descending parking functions are Dyck paths. In [3], a generalization of this result is presented that gives a bijection between descending k-Naples parking functions and a generalization of Dyck paths called k-Dyck paths. In related work by Colmenarejo et al. [4], the authors counted k-Naples parking functions through permutations, and they also defined the k-Naples area statistic. In this section, we explore when a k-Dyck path corresponds to an ascending k-Naples Parking function, giving a way of finding all k-Naples parking functions with the property that each of its rearrangements remains a k-Naples parking function. We then embed k-Dyck paths into a subset of Dyck paths to help us find other bijections with ascending and descending k-Naples parking functions.

Definition 2. A *Dyck path* of length n is a lattice path consisting of Up $(1, 1)$ and Down $(1, -1)$ steps from $(0, 0)$ to $(2n, 0)$ that never reaches below the line $y = 0$. A *k-Dyck path* of length n is also a lattice path consisting of Up and Down steps from $(0, 0)$ to $(2n, 0)$ that never reaches below the line $y = -k$ and ends with a Down step. Any such path can be represented by a sequence of U's and D's corresponding to its steps.

The following result from [3] connects k-Dyck paths to descending k-Naples parking functions.

Proposition 2 (Theorem 1.3, [3])**.** *The set of descending k-Naples parking functions of length n are in bijective correspondence with k-Dyck paths of length n.*

Remark 4. From [3], we have the following correspondence between k-Dyck paths and increasing parking preferences. A k-Dyck path P of length n uniquely corresponds to the parking preference $\alpha = (a_1, \ldots, a_n)$, where a_i equals 1 plus the

number of Down steps coming before the ith Up step. Note that α is an ascending parking preference. In [3], the descending rearrangement of α was shown to be k-Naples, and it is straightforward to reverse this process to go from decreasing k-Naples parking functions of length n to k-Dyck paths of length n.

Next, we use this correspondence between ascending parking preferences of length n and k-Dyck paths of length n to classify which k-Dyck paths correspond to ascending k-Naples parking functions.

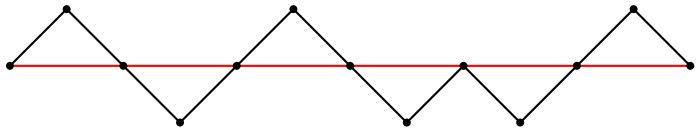

Figure 2: The k-Dyck path corresponding to $(1, 3, 3, 5, 6, 6)$.

Example 3. To go from the 2-Dyck path in Figure 2 to an ascending parking preference, we see that the first Up step has no previous Down steps making the first preference 1. The second and third Up steps then correspond to a preference of 3. Continuing in this manner yields the parking preference $(1, 3, 3, 5, 6, 6)$.

Theorem 2. *A k-Dyck path corresponds to an ascending k-Naples parking function if and only if every Down step that puts the path below the line $y = 0$ crosses back above $y = 0$ within $2k$ steps.*

Proof. First, we show that if a k-Dyck path always crosses back above the line $y = 0$ within $2k$ steps of it crossing below $y = 0$, then it corresponds to an ascending parking function. We justify this by induction on k. We know this is true for ordinary parking functions, i.e., 0-Naples parking functions, and we assume it is true for all values up to $k - 1$. Suppose we have a k-Naples parking function corresponding to a k-Dyck path that goes below the line $y = 0$ on step $2i + 1$. We may assume that the k-Naples parking function leads to the first i spots being filled by the first i cars.

By hypothesis, the path must go above the horizontal at or before step $2(i+k)+1$. If the k-Dyck path always goes above $y = 0$ before step $2(i+k)+1$ after going below $y = 0$ on step $2i + 1$, then it corresponds to an ascending $(k - 1)$-Naples parking function. So, we assume the path goes back above the horizontal for the first time on step $2(i + k) + 1$.

Now, we look at the cars c_j and their preferred parking parking spots a_j for $i + 1 \le j \le i + k + 1$. These preferences a_j correspond to Up steps below the horizontal except for a_{i+k+1}, which corresponds to the last Up step to back above the horizontal $y = 0$. For $i + 1 \le j \le i + k + 1$, parking preferences a_j satisfy $i+1 \le a_j \le i+k+1$. Since each car c_j is able to move back k spots and $j \le i+k+1$,

we see that these cars fill spots at or before spot $i+k+1$. But that implies the $k+1$ cars fill up the $k+1$ spots immediately after what was already filled. So, the first $i+k+1$ cars fill the first $i+k+1$ spots. If the path later goes below the horizontal at step $2i'+1$, then in the k-Naples parking function we can see the first i' spots are filled. We are now in a similar situation as before, so if the path goes above the horizontal at step $2(i'+k)+1$ or before, all the earlier positions will be filled. Continuing in this manner, we see that all positions are filled, so this does indeed correspond to a k-Naples parking function.

Assume for the sake of contradiction that P is a k-Dyck path corresponding to an ascending k-Naples parking function, where P crosses below the line $y=0$ and does not cross back above this line within $2k$ steps. Let step $2i+1$ be the first step where the path goes below $y=0$ but does not cross back above $y=0$ within $2k$ steps. Now, let us look at car c_{i+j} with $1 \leq j \leq k+1$. We see that c_{i+j} must have preference larger than $i+j$ since the path is below the horizontal and there are more Down than Up steps during this section. For the car to move back to spot $i+1$, all spots between $i+1$ and the parking preference, including the preference, must be filled. We see that these are spots $i+2, i+3, \ldots, i+j+1$, of which there are j. However, they could only be filled by cars $c_{i+1}, c_{1+2}, \ldots c_{i+j-1}$, of which there are $j-1$. Hence, one of these spots is open, and c_{i+j} cannot fill spot $i+1$. We see that cars c_{i+k+2} and later must have parking preference larger than or equal to $i+k+2$. This follows from the fact that the path does not go below the diagonal until at least step $2(i+k)+2$ by assumption. So no car fills spot $i+1$, showing that the path does not correspond to a k-Naples parking function. □

Corollary 1. *Every rearrangement of a parking preference is a k-Naples parking function if and only if whenever its corresponding k-Dyck path has a Down step which crosses the line $y=0$, the following $2k$ steps have a point where there have been in total two more Up steps than Down steps so far into the path.*

Proof. This follows directly from Theorems 2 and 1. □

Example 4. From Figure 2, we see that $(1,3,3,5,6,6)$ is not a 1-Naples parking function even though the lattice path is a 1-Dyck path. In fact, we can see that at step 7 it crosses below $y=0$, and then it takes four steps for the lattice path to cross back above $y=0$. Thus, it is a 2-Naples parking function.

Remark 5. In particular, for the 1-Naples case the path cannot be below the $y=0$ line for more than three steps at a time. This means that we cannot have two consecutive valleys under the $y=0$ line. This special case is equivalent to Conjecture 5 in [3].

Next, we use Proposition 2 to view descending k-Naples parking functions of length n as k-Dyck paths of the same length, and then embed these into usual Dyck

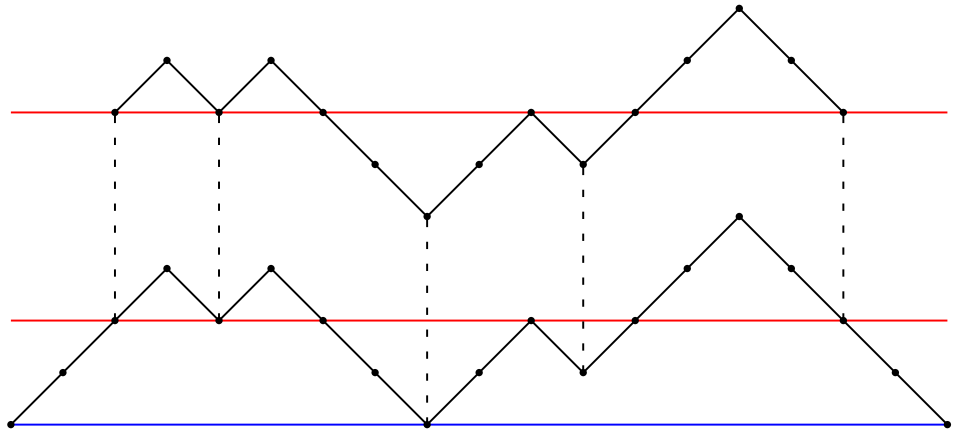

Figure 3: The k-Dyck path to Dyck path transformation.

paths of length $n + k$. This allows us to obtain many similar bijections for both ascending and descending k-Naples parking functions to other subsets of Catalan objects.

Proposition 3. *Descending k-Naples parking functions are in bijective correspondence with Dyck paths of length $n+k$ whose first k steps are Up and last $k+1$ steps are Down.*

Proof. This result follows from using the bijection between descending k-Naples functions and k-Dyck paths and then embedding these k-Dyck paths into the usual Dyck paths. Specifically, given a descending k-Naples parking function find the corresponding k-Dyck path. Then, shift this path k units right and k units up so that it starts at (k, k) and concatenate this with the lattice path of all Up steps from $(0, 0)$ to (k, k) and the lattice path of all Down steps from $(2n+k, k)$ to $(2n+2k, 0)$. In terms of Up steps and Down steps, this corresponds to appending k Up steps to the start of the k-Dyck path and k Down steps to the end of the k-Dyck path.

Moreover, reversing the process results in a k-Naples parking function. □

Remark 6. Figure 3 gives an example of this transformation for our running example $(6, 6, 6, 5, 5, 2, 1)$. Notice that for a k-Dyck path, the corresponding Dyck path represents a descending parking function of length $n + k$ that ends in at least k cars with preference 1. This leads to the following result.

Corollary 2. *Descending k-Naples parking functions of length n are in bijective correspondence with descending parking functions of length $n + k$ which end with at least k cars with preference* 1.

Similar to the transformation in Proposition 3, we see that Dyck paths of length $n+k$ that do not return to the line $y=0$ until the last step are in bijection with Dyck paths of length $n+k-1$ by removing the Up step and last Down step. This motivates the next result.

Definition 3. A parking preference is *strictly* k-Naples if it is k-Naples but not $(k-1)$-Naples.

Proposition 4. *The descending k-Naples parking functions that are not descending $(k-1)$-Naples parking functions are in bijective correspondence with Dyck paths of length $n+k$ whose first k steps are Up and last $k+1$ steps are Down and which return to the line $y=0$ sometime before the last step.*

Proof. Using the same translation between a k-Dyck path and a Dyck path as before, observe that Dyck paths that do not return to the horizontal before the last step correspond to k-Dyck paths that reach at most the line $y=-k'$ for some $0<k'<k$ and thus correspond to k'-Dyck paths. This shows our desired result. □

Finally, we may also use this embedding of k-Dyck paths into Dyck paths to see which Dyck paths are in correspondence with ascending k-Naples parking functions. The following corollary follows directly from Theorem 2.

Corollary 3. *Ascending k-Naples parking functions are in bijective correspondence with Dyck paths with length $n+k$ whose first k steps are Up, last $k+1$ steps are Down, and before the last $k+1$ steps, whenever a Down step puts the path below the line $y=k$, the following $2k$ steps have a point with two more Up than Down steps.*

4. Enumeration of Monotonic k-Naples Parking Functions

In the previous section we found bijections between either of the two types of monotonic k-Naples parking functions–descending or ascending–and subsets of Dyck paths. We now use the bijection between k-Naples parking functions and Dyck paths to give a recursive formula for the number of ascending k-Naples parking functions and also give results about the generating functions for the sequences corresponding to these objects. We end the section with closed formulas for their descending counterparts.

For the rest of this section, we fix the following notation.

- Let $I_{n,k}$ denote the number of ascending k-Naples parking functions of length n and define $I_k(x)$ to be the ordinary generating function of $I_{n,k}$.
- Let $U_{n,k}$ denote the number of ascending k-Naples parking functions of length n which start with 1 and define $U_k(x)$ to be the ordinary generating function of $I_{n,k}$.

- Let C_k denote the kth Catalan number and define $C(x)$ to be the ordinary generating function of C_k.
- Let F_{n+1} denote the $(n+1)$th Fine number (OIES sequence A000957) and define $F(x)$ to be the ordinary generating function of F_{n+1}.

Theorem 3. *For $n-1 \geq k \geq 1$ and $n \geq 0$, we have*

$$I_{n,k} = I_{n,k-1} + C_k \sum_{i=0}^{n-k} (I_{i,k-1})(U_{n-k-i,k}) \text{ and} \tag{1}$$

$$U_{n,k}x = U_{n,k-1} + C_k \sum_{i=0}^{n-k} (U_{i,k-1})(U_{n-k-i,k}). \tag{2}$$

Remark 7. Note that $I_{n,0} = C_n$ and $U_{0,k} = 0$, otherwise $U_{n,0} = C_n$. Further observe that the k-Naples parking functions which start with 1 correspond to k-Dyck paths that start with an Up step. In Theorem 3, if $n \leq k$, then both summations are empty making them 0. This corresponds to there being no new k-Naples for a fixed length n if k is large enough.

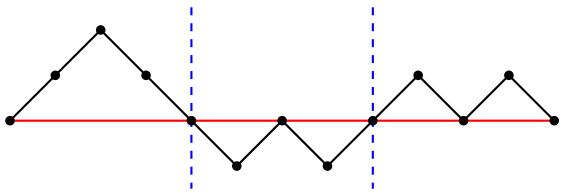

Figure 4: Breakdown for the recurrence for ascending k-Naples parking functions.

Proof. For $I_{n,k}$, we need to find the new ascending k-Naples parking functions which are not represented in $I_{n,k-1}$ and add the two together. Recall that $I_{n,0} = C_n$ when $n > 0$, giving the base for our recurrence. From Theorem 2, we know that an ascending k-Naples parking function of length n that is not a $(k-1)$-Naples parking functions must have a corresponding Dyck path that is below the horizontal for exactly $2k$ steps. Let there be $2i$ steps before the point it goes below the horizontal for $2k$ steps. We see that i of these are Up steps since the last step must be to the horizontal. Also, notice that the last step is a Down step as otherwise the path would be below the horizontal for at least $2k+2$ steps. So, the number of ways to get to this point is the number of $(k-1)$-Naples parking functions of length i, recalling that the last step of the k-Dyck paths corresponding to ascending Naples parking functions must be Down. So, we see that there are $I_{i,k-1}$ such paths. This argument corresponds to the first section in Figure 4.

Now, once the path has gone below the horizontal, it must stay there for $2k$ steps. At this point, it must return to the horizontal. Flipping this across the horizontal gives regular Dyck paths of length k, leading to C_k possibilities. This argument corresponds to the second section in Figure 4.

Now, the path is at the horizontal after $2(i+k)$ steps. It must then go up so as to not stay below the horizontal for too long. But we see that the final section of length $n-i-k$ starting with an Up step has $U_{n-i-k,k}$ options, which can be seen in the third section of Figure 4.

Summing over all i gives Equation (1) and Equation (2) is proved similarly. □

Given the recursive formulas in Theorem 3, we note a connection between the 1-Naples paths that start with an Up step and the Fine Numbers, an integer sequence closely related to the Catalan Numbers. Many interpretations of the Fine Number sequence can be found in [5]. We then proceed to present more general recursive formulas for the associated ordinary generating functions of $I_{n,k}$ and $U_{n,k}$.

Theorem 4. *For $n \geq 0$, we have $U_{n,1} = F_{n+1}$, where F_{n+1} denotes the $(n+1)$th Fine Number (OIES sequence A000957).*

Proof. It suffices to prove that $U_1(x) = \frac{F(x)-1}{x}$ (excluding F_1 and reindexing). We know that $U_{n,0} = C_n$ for all $n > 0$, and $U_{0,0} = 0 = C_0 - 1$; so we have $U_0(x) = C(x) - 1$. Now, for any given $n > 0$ (and $k = 1$), from (2) we have

$$x^n U_{n,1} = x^n U_{n,0} + x \sum_{i=0}^{n-1} x^i U_{i,0} x^{n-i-1} U_{n-i-1,1}.$$

Adding these equations for all $n > 0$ we get

$$\sum_{i=1}^{\infty} x^n U_{n,1} = \sum_{i=1}^{\infty} x^n U_{n,0} + x \sum_{i=1}^{\infty} x^{n-1} \sum_{i=0}^{n-1} U_{i,0} U_{n-i-1,1}.$$

Noticing that $\sum_{i=0}^{n-1} x^i U_{i,0} x^{n-i-1} U_{n-i-1,1}$ is the $(n-1)$th term in the convolution of $U_{n,0}$ and $U_{n,1}$, that is, the coefficient of x^{n-1} in $U_1(x)U_0(x)$, we obtain that $U_1(x) = \frac{C(x)-1}{1+x-xC(x)}$. But we know that $F(x) = \frac{1}{1-x^2C^2(x)}$, and that $xC^2(x) - C(x) + 1 = 0$, so

$$U_1(x) = \frac{C(x)-1}{1+x-xC(x)} = \frac{F(x)}{x} - \frac{1}{x} = \frac{F(x)-1}{x}.$$ □

Theorem 5. *For $n \geq 0$, $I_{n,1} = CF_n$, where CF is the convolution of the Catalan numbers with the Fine numbers (OIES sequence A000958).*

Proof. Let $I_1(x)$ be the ordinary generating function for $I_{n,1}$ and $I_0(x)$ be the one for $I_{n,0}$. Since $I_{0,1} = I_{0,0}$ and $I_0(x) = C(x)$ we get, by a very similar argument as Theorem 4,

$$I_1(x) = I_0(x) + xI_0(x)U_1(x) = C(x) + xC(x)\frac{F(x)-1}{x} = C(x)F(x).$$ □

Theorem 6. *Let $U_k(x)$ represent the ordinary generating function for $U_{k,1}$, and define $U_{k-1}(x)$, $I_k(x)$, and $I_{k-1}(x)$ similarly. Then*

$$I_k(x) = I_{k-1}(x) + C_k x^k I_{k-1}(x) U_k(x) \text{ and} \tag{3}$$

$$U_k(x) = U_{k-1}(x) + C_k x^k U_{k-1}(x) U_k(x). \tag{4}$$

Proof. The idea of the proof is identical to Theorem 5, except for a given k we are only able to use recurrence relations for $n \geq k$, otherwise the convolution sum becomes meaningless. This means that we are never adding the coefficients representing degrees smaller than k for both $I_k(x)$ and $I_{k-1}(x)$ in Equation (3) and conversely for U in Equation (4). But this is not an issue, since for $n < k$ any ascending parking preference is $(k-1)$-Naples (and thus also k-Naples). so the coefficients are the same and adding them on both sides of the equation do not change the result, so we can use the exact same reasoning as for Theorem 4. □

For $k \geq 2$, the generating functions become increasingly cumbersome to work with towards finding a closed formula. Using our bijection between k-Naples parking functions and Dyck paths, we can obtain a closed formula for the number of descending k-Naples parking functions and descending strictly k-Naples parking functions.

Theorem 7. *The total number of descending k-Naples parking functions of length n is*

$$\binom{2n-1}{n} - \binom{2n-1}{n+k+1}$$

and the number of descending strictly k-Naples parking functions of length n is

$$\frac{k+1}{n}\binom{2n}{n+k+1}.$$

Proof. We first seek to compute the number of descending k-Naples parking functions of length n. Using the bijective correspondence of Proposition 2, these correspond to the lattice paths from $(0,0)$ to $(2n,0)$ with last step Down that never go below the line $y = -k$. We can first compute the number of lattice paths from $(0,0)$ to $(2n-1,1)$ that never go below the line $y = -k$. Using a reflection with respect to the line $y = -k-1$, we obtain that the number of lattice paths from $(0,0)$ to $(2n-1,1)$ minus the number of lattice paths from $(0,0)$ to $(2n-1,-2k-3)$ is exactly the desired result.

To obtain the number of descending strictly k-Naples parking functions we subtract the the total number of $(k-1)$-Naples parking functions, which is

$$\left(\binom{2n-1}{n} - \binom{2n-1}{n+k+1}\right) - \left(\binom{2n-1}{n} - \binom{2n-1}{n+k}\right).$$

Simplifying this expression yields our desired result. □

5. Other Bijections

In the previous sections, we found a bijection between ascending and descending k-Naples parking functions of length n and subsets of Dyck paths of length $n+k$. Considering there are well-known bijections between Dyck paths, full binary trees, triangulations of an $(n+2)$-gon, and non-crossing partitions of $[n]$ (see [8], for example), we describe which subsets of these objects are in bijection with descending strictly k-Naples parking functions. Since these correspondences follow directly from the bijections between descending parking functions and the various Catalan objects, we omit these proofs and provide an illustration in Figure 5 for the descending strictly 2-Naples parking function $(6,6,6,5,5,2,1)$.

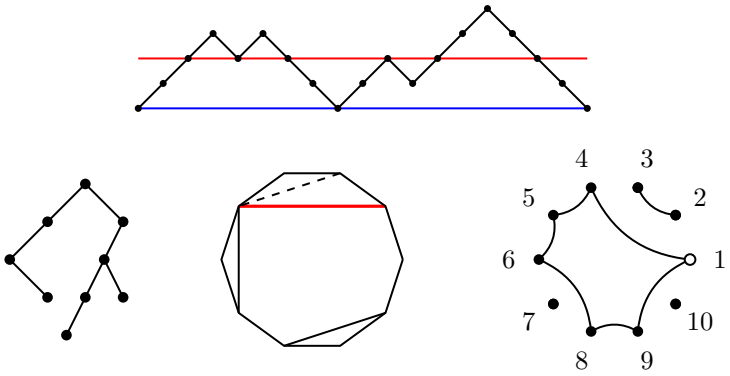

Figure 5: The Catalan objects corresponding to the descending strictly 2-Naples parking function $(6,6,6,5,5,2,1)$.

Proposition 5. *Descending strictly k-Naples parking functions are in bijection with binary trees that have $n+k$ nodes and satisfy the properties that the root has at least $k-1$ left children in a row, has a right child, and this right child has at least k left children in a row.*

Definition 4. An *r-in-s dissection* is a dissection of an s-gon into an r-gon and $(s-r)$ triangles.

Definition 5. An *r-rooted non-crossing set partition* is a non-crossing set partition where one of the parts, the root, has size r.

Proposition 6. *Descending strictly k-Naples parking functions of length n are in bijection with $(2k+2)$-in-$(n+k+1)$ dissections, up to rotation, but with a distinguished edge on the $(2k+2)$-gon.*

Proposition 7. *Descending strictly k-Naples parking functions of length n are in bijection with $(2k+2)$-rooted non-crossing partitions of $[n+k+1]$, where 1 is in the root.*

6. Future Work

As mentioned previously, all rearrangements of parking functions are still parking functions. Using this fact, it is possible to find simple labeling rules on Dyck paths and trees that correspond to every parking function [9]. One area of future research is to explore a way of describing which rearrangements of descending k-Naples parking functions are still k-Naples parking functions based on a labeling of the objects with which they are in bijection.

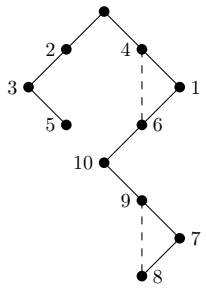

Figure 6: Labeling binary trees for $k = 1$.

When $k = 1$, we noticed some patterns when labeling trees that correspond to a 1-Naples parking function as illustrated in Figure 6. Note that in these labelings, the root is unlabeled as it does not correspond to a car, and a node must have a lower labeling than its right child. It can be observed that if a direct right descendent of the root does not have a left child, then it must have a higher labeling than its right child. Lastly, we consider the final direct right descendent in the section without a left child. It must have a higher label than its right child, so we consider the right child that is connected to the original node, depicted in Figure 6 by a dotted line. If the grandchild has a smaller label than the original, there is no problem with the rearrangement. Otherwise, the process must begin again. This is a rather convoluted process, and neither of the rules follow in a satisfying way once $k > 1$.

A similar direction is to find what labeling conventions correspond to ascending k-Naples parking functions using our other bijections. We have seen the result for both Dyck paths and binary trees, but we have not studied the condition on dissections or rooted non-crossing partitions. Perhaps one of these settings could better help us understand rearrangements.

Lastly, there are many objects counted by the Catalan numbers and their convolutions that we have not discussed here. Finding and understanding more bijections could help us better understand the structure of k-Naples parking functions, and some objects may be better suited for describing rearrangements. One could also look for bijections for ascending strictly k-Naples parking functions, ascending or descending parking preferences that are not k-Naples parking functions, or descend-

ing k-Naples parking functions whose ascending rearrangements are not k-Naples parking functions.

Acknowledgements. Part of this research was performed with support from the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1440415). Pamela E. Harris was supported through a Karen EDGE Fellowship. Andrés R. Vindas-Meléndez was partially supported by the National Science Foundation under Awards DGE-1247392, KY-WV LSAMP Bridge to Doctorate HRD-2004710, and DMS-2102921.

References

[1] A. Baumgardner, *The Naples Parking Function*, Honors Contract-Graph Theory, Florida Gulf Coast University, 2019.

[2] J. Carlson, A. Christensen, P. E. Harris, Z. Jones, and A. Ramos Rodriguez, *Parking functions: choose your own adventure*, College Math. J. **52** (4) (2021), 254-264.

[3] A. Christensen, P. E. Harris, Z. Jones, M. Loving, A. Ramos Rodríguez, J. Rennie, and G. Rojas Kirby, *A generalization of parking functions allowing backward movement*, Electron. J. Combin. **27** (1) (2020), P1.33.

[4] L. Colmenarejo, P. E. Harris, Z. Jones, C. Keller, A. Ramos Rodríguez, E. Sukarto, and A. R. Vindas-Meléndez, *Counting k-Naples parking functions through permutations and the k-Naples area statistic*, Enumer. Comb. Appl. **1** (2) (2021), #S2R11.

[5] E. Deutsch and L. Shapiro, *A survey of the Fine numbers* Selected papers in honor of Helge Tverberg, Discrete Math. **241** (1-3) (2001), 241-265.

[6] A. G. Konheim and B. Weiss, An Occupancy Discipline and Applications, *SIAM J. Appl. Math.* **14** (6) (1966), 1266-1274.

[7] R. Pyke, The supremum and infimum of the Poisson process, *J Ann. Math. Statist.* **30** (2) (1959), 568-576.

[8] R. P. Stanley, *Catalan numbers*, Cambridge University Press, New York, 2015.

[9] C. H. Yan, Parking functions, in *Handbook of enumerative combinatorics*, CRC Press, Boca Raton, FL, 2015.