Journal of Integer Sequences, Vol. 21 (2018),
Article 18.2.3

Descents in Parking Functions

Paul R. F. Schumacher
1512 Oakview Street
Bryan, TX 77802
USA
Paul.R.F.Schumacher@gmail.com

Abstract

Parking functions are a superset of permutations. In this article, we count the number of parking functions of length n with a fixed number of ties, as well as the number of descents in those parking functions. The steps to achieve this result also reveal many things about their internal structure.

1 Introduction

Parking functions were introduced in 1966 by Konheim and Weiss [4]. 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. We refer to the set of parking functions of length n as PF_n.

Definition 1. Throughout, we use the notation $[n]$ to mean the set of integers $\{1, 2, \ldots, n\}$.

Definition 2. Let a be a map from $[n]$ to $[n]$ (written $(a_1\, a_2 \cdots a_n)$) such that for all $i \leq n$, the number of j such that $a_j \leq i$ is greater than or equal to i. (Alternatively, if b_i are the a_i sorted into non-decreasing order, then $b_i \leq i$.) A map that satisfies this property is called a *parking function* of length n [4].

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. The authors proved that the map from parking functions to Prüfer codes was a bijection between PF_n and $[n+1]^{n-1}$ (proof omitted for brevity), giving us an alternate method of counting $|\,\mathrm{PF}_n\,| = (n+1)^{(n-1)}$.

Definition 3. For a parking function $f = (a_1 a_2 \cdots a_n) \in \mathrm{PF}_n$, define the *Prüfer code* of f to be $((a_2 - a_1) \bmod (n+1), \ldots, (a_n - a_{n-1}) \bmod (n+1))$. See [3].

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

Definition 4. 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 $x = y$.) A *peak* in a Dyck path is an up step followed by an over step [5].

Definition 5. 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 6. $\mathrm{PF}_n \cong \mathrm{LD}_n$.

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

Given a labeled Dyck path, let d_i be the number of up steps before the ith 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 are 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 $\mathrm{PF}_n \to \mathrm{LD}_n$ and exhibited an inverse. (Proof paraphrased from Loehr [5], given here to exhibit the map we use later. See Figure 1 for an illustrative example.) □

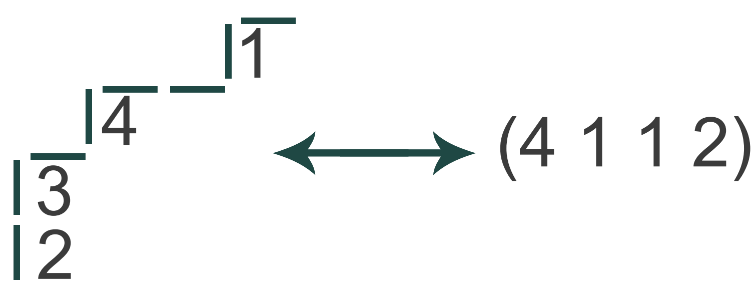

Figure 1: Labelled Dyck path to parking function example.

Let IP_n be the set of non-decreasing parking functions, i.e., those parking functions $(a_1\, a_2 \cdots a_n)$ 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 $2n$.

Corollary 7. $\mathrm{IP}_n \cong D_n$.

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. □

2 Descents in parking functions

2.1 Counting descents

Definition 8. Given a parking function $(a_1\, a_2 \cdots a_n)$, we call a pair $(a_i,\, a_{i+1})$ 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.

Definition 9. Let $\mathrm{PF}_{(n,i)} \subset \mathrm{PF}_n$ 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 2) 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.

1
15 15
50 200 50
50 1030 1030 50
15 1240 3970 1240 15
1 407 3480 3480 407 1

Figure 2: Parking function distribution for $n = 6$

Theorem 10. *There are* $\binom{n}{2}(n+1)^{n-2}$ *total descents among all parking functions in* PF_n*.*

In order to prove this result, we will take small steps exploring the internal distribution of the ties, ascents, and descents in PF_n, using counting arguments.

Lemma 11. *If* $\mathrm{IP}_{(n,i)}$ *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* $\mathrm{IP}_{(n,i)} \cong D_{(n,n-i)}$.

Proof. Given any non-decreasing parking function with i ties, we note that there must be exactly $n-i$ different elements in the parking function. When we use the map from (6) to a Dyck path, we get a Dyck path with exactly $n-i$ peaks. □

Lemma 12. *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. The number of Dyck paths with exactly j peaks is given by the triangle of Narayana numbers A001263, $T(n,j) = \frac{1}{j}\binom{n}{j-1}\binom{n-1}{j-1}$. 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 see that

$$\frac{1}{n-i}\binom{n-1}{n-i-1}\binom{n}{n-i-1} = \frac{1}{i+1}\binom{n-1}{i}\binom{n}{i}.$$

□

Theorem 13. *There are* $\binom{n-1}{i}n^{n-1-i}$ *parking functions in* $\mathrm{PF}_{(n,i)}$.

Proof. Using the bijection in Definition 3, we note that each tie in a parking function becomes a 0 in the corresponding Prüfer code, and that any sequence of $(b_1 b_2 \cdots b_{n-1})$ in $[n+1]_0^{n-1}$ 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 elements of $[n]$ (nonzero elements of $[n+1]_0$). 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 $\mathrm{PF}_{(n,i)}$. □

Remark 14. This proof was given by the author in his dissertation in 2009 [6, Thm A.4]. The result can also be derived from the q-nomial formula given by Yan in 2015 [1, Corollary 1.3 in Chapter 13].

Lemma 15. *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 13 with $j = n-1-i$. □

Lemma 16. *The distribution of ascents and descents in* PF_n *and* $\mathrm{PF}_{(n,i)}$ *are symmetrical. That is, the total descents among the parking functions in either set is the same as the total ascents among the parking functions in that set.*

Proof. If we flip a parking function and look at $(a_n a_{n-1} \cdots a_1)$, 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 $\mathrm{PF}_{(n,i)}$ of PF_n, meaning that the number of descents in $\mathrm{PF}_{(n,i)}$ must equal the number of ascents. □

Lemma 17. *There are* $\frac{n-1-i}{2}\binom{n-1}{i}n^{n-1-i}$ *total descents among all parking functions in* $\mathrm{PF}_{(n,i)}$.

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

Theorem 18. *If* a_i *is the number of descents in parking functions of length* n *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 Lemma 17, we know that there are $\frac{n-1-i}{2}\binom{n-1}{i}n^{n-1-i}$ descents in $\mathrm{PF}_{(n,i)}$, so we sum this number over i, giving

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

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

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

□

Setting $y = 1$ completes our proof of Theorem 10.

Corollary 19. *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. Of these, $\binom{n}{2}(n+1)^{n-2}$ 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. □

By symmetry, all of these results hold for ascents.

Remark 20. This proof was given by the author in his dissertation in 2009 [6, Corollary A.12]. A probabilistic argument was given in 2017 by Diaconis and Hicks in [2, Eq. 4.1].

3 Tree implications

Two related results for trees are implied by the relationship of trees to Prüfer codes.

Theorem 21. $\mathrm{PF}_n \cong \mathrm{LT}_{n+1}$*, 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 with vertices 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 $[n+1]_0$, 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. (Proof paraphrased from Foata and Riordan [3], repeated here to show the map we use below.) □

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.

Corollary 22. *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 Theorem 13. □

Corollary 23. *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, and then re-root the tree at 0. 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 $i+1$. □

4 Remaining questions

This article has explored the structure of the steps of parking functions and given formulas for several aspects of that structure, but there are questions remaining to be explored further.

If we count the number of parking functions in PF_n with i ties and j descents (leaving $n-1-j-i$ ascents), and arrange the totals in a grid (see Figure 2), several interesting properties appear. The numbers along the diagonal edges are the Narayana numbers (as shown in Theorem 12). The row sums are given in Theorem 13. However, a general formula which gives the individual elements within each row (for rows of length ≥ 3) is, as yet unknown. The elements within the rows show no nice properties, nor are they given by any sequence in the OEIS. Investigation into these elements shows that the inner terms are given by summation of multiples of earlier elements in the table. This can be seen by creating PF_n from $\mathrm{PF}_{i<n}$ via insertion of new elements into smaller parking functions. A general formula for these sums would require finding a recursion algorithm between PF_n and $\mathrm{PF}_{i<n}$ which preserves step identities (ascent, descent, or tie).

References

[1] M. Bóna, editor, *Handbook of Enumerative Combinatorics*, Chapman and Hall/CRC, 2015.

[2] P. Diaconis and A. Hicks, Probabilizing parking functions, *Adv. in Appl. Math* **89** (2017), 125–155.

[3] A.D. Foata and J. Riordan, Mappings of acyclic and parking functions, *Aequationes Math.* **10** (1974), 10–22.

[4] A. G. Konheim and B. Weiss, An occupancy discipline and applications, *SIAM J. Appl. Math.* **14** (1966), 1266–1274.

[5] N. A. Loehr, Combinatorics of q,t-parking functions, *Adv. in Appl. Math.* **34** (2005), 408–425.

[6] P. R. F. Schumacher, *Parking Functions and Generalized Catalan Numbers*, PhD thesis, Texas A&M University, 2009.

2010 *Mathematics Subject Classification*: Primary 05A15.
Keywords: parking function, descent, Dyck path.

(Concerned with sequence A001263.)

Received November 15 2017; revised versions received February 24 2018; February 26 2018; March 2 2018. Published in *Journal of Integer Sequences*, March 3 2018.

Return to Journal of Integer Sequences home page.