Prime parking functions on rooted trees

Westin King, Catherine H. Yan

Department of Mathematics, Texas A&M University, College Station, TX 77843, United States

A R T I C L E   I N F O

A B S T R A C T

For a labeled, rooted tree with edges oriented towards the root, we consider the vertices as parking spots and the edge orientation as a one-way street. Each driver, starting with her preferred parking spot, searches for and parks in the first unoccupied spot along the directed path to the root. If all n drivers park, the sequence of spot preferences is called a parking function. We consider the sequences, called *prime* parking functions, for which each driver parks and each edge in the tree is traversed by some driver after failing to park at her preferred spot. We prove that the total number of prime parking functions on trees with n vertices is $(2n-2)!$. Additionally, we generalize *increasing* parking functions, those in which the drivers park with a weakly-increasing order of preference, to trees and prove that the total number of increasing prime parking functions on trees with n vertices is $(n-1)!S_{n-1}$, where $\{S_i\}_{i \geq 0}$ are the large Schröder numbers.
© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Konheim and Weiss first studied classical parking functions in 1966 by examining the probability that a random hashing function would successfully store data when collisions

---

E-mail addresses: wking@math.tamu.edu (W. King), cyan@math.tamu.edu (C.H. Yan).

Fig. 1. A path of 5 parking spots.

were resolved via linear probing [7]. Since then, parking functions have appeared in the study of many other combinatorial objects such as noncrossing partitions, hyperplane arrangements, posets, and trees (see [11] and [12]). A straightforward description of a classical parking function $s \in [n]^n$ is that of n drivers, each with a preferred parking spot s_i, wishing to park on a one-way street with parking spots labeled 1 to n. One-by-one the drivers attempt to park using the following parking procedure:

1. Driver i parks at s_i if it is available.
2. If it is not, she parks at the first available spot after s_i.
3. If there are none, she leaves the parking lot without parking.

Definition 1. The sequence of parking preferences $s = (s_1, s_2, \ldots, s_n)$ is called a *classical parking function of length n* if the parking procedure results in all n drivers successfully parking.

The sequences $(1, 3, 2, 3, 1)$ and $(1, 3, 4, 4, 1)$ are parking functions in Fig. 1. The sequence $(3, 3, 3, 4, 5)$ is not, as the spots 1 and 2 are unoccupied after drivers attempt to park. The total number of classical parking functions of length n is $(n + 1)^{(n-1)}$ [7].

It is well-known that a permutation of a classical parking function is still a classical parking function. That is, if $s \in [n]^n$ is a classical parking function and $\sigma \in \mathfrak{S}_n$, then $(s_{\sigma(1)}, s_{\sigma(2)}, \ldots, s_{\sigma(n)})$ is also a parking function. Those parking functions for which the s_i's are weakly increasing, called *increasing parking functions*, are counted by the ubiquitous Catalan numbers $\{\frac{1}{n+1}\binom{2n}{n}\}_{n \geq 0}$. Rearranging the above examples, $(1, 1, 2, 3, 3)$ and $(1, 1, 3, 4, 4)$ are both classical increasing parking functions.

Another interesting subset of classical parking functions is defined by the parking functions which, after any 1 is removed from the sequence, are classical parking functions on the first $n - 1$ spots. Such parking functions were called *prime* by Gessel and their count is given by $(n - 1)^{(n-1)}$ ([11], Exercise 5.49). For example, $(1, 3, 2, 3, 1)$ is a prime parking function as both $(3, 2, 3, 1)$ and $(1, 3, 2, 3)$ are parking functions on 4 spots. Notice that any reordering of a prime parking function is also a prime parking function, since rearranging then removing a 1 is the same as removing a 1 and rearranging appropriately.

Many generalizations of parking functions exist, including rational parking functions [1], G-parking functions [10], $\vec{u}$-parking functions [8], and parking sequences involving cars of different lengths [4]. The particular generalization this paper concerns itself with extends the process by which drivers find a place to park and is due to Lackner and Panholzer [9]. Let T be a rooted tree with vertex set $[n]$ and edges oriented towards the root. In this case, we will denote $|T| := n$. Like the classical case, let $s \in [n]^n$ be the sequence of preferred spots. One by one, driver i will attempt to park at s_i. If it is already occupied, the driver travels along the unique path towards the root, parking in
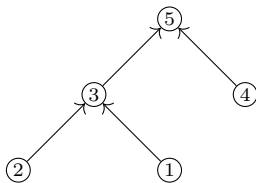
Fig. 2. A tree with parking function $(2, 2, 1, 4, 2)$.

the first available spot. If no spots are available, then the driver leaves the tree without parking.

Definition 2. A sequence $s \in [n]^n$ is a *parking function on T* if the parking procedure allows all cars to park on T.

We call the pair (T, s) a *parking function* and give an example in Fig. 2. Notice that tree parking functions are classical parking functions when T is a path with vertices labeled in decreasing order away from root n. We denote by $\mathcal{P}_n$ the path with n vertices on which classical parking functions are defined. Fig. 1 is a picture of $\mathcal{P}_5$. We also note that, because vertices have outdegree at most one, the parking procedure is deterministic on rooted trees and thus well-defined.

Let F_n be the total number of pairs (T, s) for which $|T| = n$ and set $F(x) = \sum_{n \geq 1} F_n \dfrac{x^n}{(n!)^2}$. Lackner and Panholzer [9] found that $F(x)$ satisfies the equation

$$F(x) = T(2x) + \ln\left(1 - \frac{T(2x)}{2}\right), \tag{1}$$

where $T(x) = \sum_{n \geq 1} n^{n-1} \dfrac{x^n}{n!}$ is the tree function. Furthermore, they calculated

$$F_n = ((n-1)!)^2 \cdot \left(\sum_{i=0}^{n-1} \frac{(n-i) \cdot (2n)^i}{i!}\right).$$

In Section 2, we define a generalization of classical prime parking functions to trees. In Section 3, we decompose a parking function (T, s) and use Equation (1) to prove the following theorem.

Theorem 1. *The total number of prime tree parking functions P_n for $n \geq 1$ is given by*

$$P_n = (2n - 2)!$$

In Section 4, we give a bijective proof for Theorem 1 by matching a prime parking function (T, p) with a permutation and a rooted, ordered (plane) tree with labeled non-

root vertices. In, Section 5 we count the number of prime parking functions (T, p) by the number of lucky drivers, those who park in their preferred parking space, to find

Theorem 2. *When $n \geq 2$, the number of prime parking functions, (T, p), with $|T| = n$ and k lucky drivers is given by*

$$n!(n-1)!N_{n-1,k},$$

where $N_{n-1,k} = \frac{1}{n-1}\binom{n-1}{k}\binom{n-1}{k-1}$, a Narayana number.

In Section 6, we consider special subsets of the domain and image of a bijection constructed in Section 4. In Section 7, we consider a natural generalization of increasing parking functions, called *parking distributions*, first studied by Butler, Graham, and Yan [3].

Definition 3. A *parking distribution* is a parking function (T, s) such that s is weakly increasing.

Like the classical case, the order in which drivers attempt to park does not affect their ability to park. As such, the name "distribution" is chosen because the distribution of drivers, $|\{j : s_j = i\}|$ for $i \in V(T)$, determines whether all drivers successfully park. In this paper, we let s be weakly increasing for convenience. We discuss the relationship between regular and prime parking distributions and give a generating function proof of the following theorem.

Theorem 3. *The total number of prime parking distributions on trees with $n \geq 1$ vertices is given by*

$$\widetilde{P}_n = (n-1)!S_{n-1},$$

where $\{S_i\}_{i \geq 0}$ are the large Schröder numbers.

Finally, in Section 8, we discuss some directions for further research. This paper is the full version of Section 3 of [6].

2. Prime parking functions

Let T be a rooted tree with vertex set $[n]$ and $v \in [n]$. We define a preorder $\preceq_T$ on $[n]$ by letting $v \preceq_T w$ if there exists a directed path from v to w in T. By convention we say $v \preceq_T v$. Let $v \in [n]$ and define T_v to be the subgraph induced by the set of vertices $\{u : u \preceq_T v\}$. The following characterization of a parking function is a slight modification of that given by Lackner and Panholzer [9].
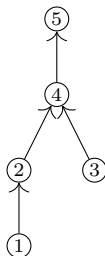
Fig. 3. T with prime $p = (1, 3, 2, 3, 1)$.

Proposition 1. *The pair (T, s) is a parking function if and only if for all $v \in [n]$, we have $|T_v| \leq |\{i : s_i \in T_v\}|$.*

Proof. Suppose (T, s) is a parking function. Then at the end of parking, every vertex is occupied by some car. For any v, the only drivers that can park in T_v are those preferring a spot in T_v, so we must have $|T_v| \leq |\{i : s_i \in T_v\}|$.

On the other hand, suppose that (T, s) is not a parking function. Then there is some driver who does not park, meaning there is some unoccupied spot v. Then $|T_v| > |\{i : s_i \in T_v\}|$, since v is empty and drivers must park in the first unoccupied spot they reach. $\square$

Notice in the classical parking function case, $(\mathcal{P}_n, s)$ is a parking function if and only if for $k \in [n]$, we have $k \leq |\{i : s_i \leq k\}|$. In addition, a classical parking function s is prime if and only if $k < |\{i : s_i \leq k\}|$ when $k \leq n - 1$ (see [11], Exercise 5.49f). In this spirit, we define prime parking functions on trees and give an example in Fig. 3.

Definition 4. A parking function (T, s) is *prime* if, for every non-root vertex $v \in [n]$, we have $|T_v| < |\{i : s_i \in T_v\}|$.

We briefly introduce some notation. If there is an edge of the form $u \to v$, we represent the edge as the ordered pair (u, v). Additionally, we denote

$$\mathcal{PF}_n = \{(T, p) : |T| = n \text{ and } p \text{ is a prime parking function on } T\},$$

and so $P_n = |\mathcal{PF}_n|$. Prime parking functions can also be understood in terms of edges crossed by cars failing to park at their preferred spots.

Definition 5. For a parking function (T, s), we say that an edge e is *used* by s if there exists some driver who, after failing to park at her preferred spot, crosses e during her search for an unoccupied spot.

Fig. 4 shows a classical parking function with both used and unused edges. Whether or not an edge (u, v) is used by a parking function depends only on the number of cars preferring T_v.
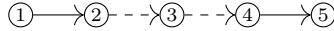
Fig. 4. Solid edges are used by $s = (1, 3, 4, 4, 1)$.

Proposition 2. *Given a parking function (T, s), an edge $e = (u, v)$ is used by s if and only if $|T_u| < |\{i : s_i \in T_u\}|$. Furthermore, the set of edges used by s is invariant under permutations of s.*

Proof. Suppose $e = (u, v)$ is used by s. Then at least one driver preferring T_u does not park in T_u. No cars preferring a vertex outside T_u can park inside, as T_u consists of all vertices w such that $w \preceq_T u$. The pair (T, s) is a parking function, so it follows that $|T_u| < |\{i : s_i \in T_u\}|$. On the other hand, if $|T_u| < |\{i : s_i \in T_u\}|$, then as s is a parking function on T, at least one driver preferring T_u must park outside. This driver must cross e in order to do so.

The set characterization of used edges does not change if the letters of s are permuted.   □

As an immediate corollary we see,

Corollary 1. *A parking function (T, p) is prime if and only if every edge in T is used by p.*

3. Proof of Theorem 1 via generating functions

Recall that F_n is the number of pairs (T, s) with $|T| = n$ and s a parking function on T, while P_n is the number of such pairs where s is prime. We define for both F_n and P_n the generating functions

$$F(x) = \sum_{n \geq 1} F_n \frac{x^n}{(n!)^2} \qquad \text{and} \qquad P(x) = \sum_{n \geq 1} P_n \frac{x^n}{(n!)^2}.$$

We choose $(n!)^2$ for the denominator to account for both the relabellings of the tree's vertices and the re-orderings of the preference sequence s.

For a parking function (T, s), we consider a decomposition of T into a "core" component supporting a prime parking function, $(T_0, s^{(0)})$, and some collection of general parking functions, $(T_i, s^{(i)})$, attached to the core component. Let T_0 be the subtree of T containing the root and all vertices connected to the root via edges used by s. Let $s^{(0)}$ be the subsequence of s defined by drivers preferring vertices in T_0. Notice that $s^{(0)}$ is a prime parking function by construction. The other subtrees T_i are the connected components remaining after deleting edges (u, v) unused by s, where $v \in V(T_0)$ and $u \notin V(T_0)$, and $s^{(i)}$ are the subsequences of s consisting of drivers preferring vertices in T_i.

Fig. 5 gives a general overview of this decomposition. Dashed edges are those unused by s but connected to the "core" component T_0. In Fig. 4, the "core" component is the
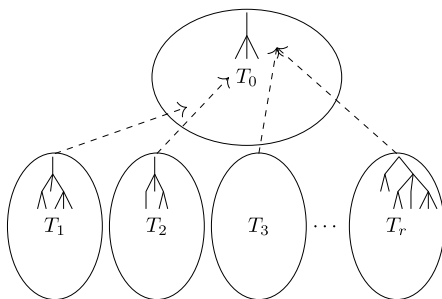
Fig. 5. Decomposition into components.

subtree induced by vertex set $\{4, 5\}$ while $s^{(0)} = (4, 4)$. The one other component is the subgraph induced by vertices $\{1, 2, 3\}$ (identical to $\mathcal{P}_3$) with $s^{(1)} = (1, 3, 1)$. Notice that $(1, 3, 1)$ is not a prime parking function on $\mathcal{P}_3$.

In this way, we can construct any parking function (T, s) with $|T| = n$ by choosing a prime parking function $(T_0, s^{(0)})$ with $|T_0| = k_0$ and r-many other regular parking functions $\{(T_i, s^{(i)})\}_{i=1}^r$ with $k_i = |T_i| \geq 1$ and $\sum_{i=0}^r k_i = n$. From there, we can attach each T_i to T_0 in one of k_0-many places to form T. What remains is to choose the labels on T and choose which indices in s each $s^{(i)}$ is assigned. The $1/r!$ accounts for the order in which the T_i are chosen and attached. This means

$$F_n = \sum_{r \geq 0} \frac{1}{r!} \sum_{\sum_{i=0}^r k_i = n} P_{k_0} F_{k_1} \cdots F_{k_r} \binom{n}{k_0, k_1, \ldots, k_r}^2 (k_0)^r.$$

Since $F_1 = 1$, summing over $n \geq 1$, we get the relationship

$$F(x) = P\left(xe^{F(x)}\right). \tag{2}$$

Using (1),

$$P\left(xe^{F(x)}\right) = T(2x) + \ln\left(1 - \frac{T(2x)}{2}\right). \tag{3}$$

Setting $z = z(x) = xe^{F(x)}$, $y = y(x) = \frac{T(2x)}{2}$, and using the relation $T(x) = xe^{T(x)}$, we notice from Equation (1) that

$$z = y(1 - y).$$

Solving the quadratic equation gives

$$y = zC(z),$$

where $C(x) = \sum\limits_{n \geq 0} C_n x^n$ is the ordinary generating function for the Catalan numbers with analytic expression

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

Since $z(0) = 0$ and $z_1 \neq 0$, the formal power series $z(x)$ has a compositional inverse. Rewriting Equation (3) and plugging in the inverse, we get

$$P(x) = 2xC(x) + \ln(1 - xC(x)).$$

The Catalan generating function $C(x)$ satisfies the recursion $C(x) = 1 + xC(x)^2$ and thus $C'(x)$ satisfies

$$C'(x) = \frac{C(x)^2}{1 - 2xC(x)}.$$

After some algebra, we can see

$$\frac{C(x)}{(xC(x))'} = \frac{1 - 2xC(x)}{1 - xC(x)}. \tag{4}$$

Then taking the derivative of $P(x)$ and using (4), we have

$$
\begin{aligned}
P'(x) &= 2(xC(x))' - \frac{(xC(x))'}{1 - xC(x)} \\
&= (xC(x))' \left(\frac{1 - 2xC(x)}{1 - xC(x)} \right) \\
&= C(x).
\end{aligned}
$$

Therefore,

$$P(x) = \sum_{n \geq 1} \frac{C_{n-1}}{n} x^n = \sum_{n \geq 1} (2n - 2)! \frac{x^n}{(n!)^2}.$$

Hence, $P_n = (2n - 2)!$ as claimed. Such a simple number demands a bijective proof, which we give in the next section.

4. Bijective proof of Theorem 1

In order to determine P_n, we define a bijection $\psi : (T, p) \mapsto (\sigma, P)$ where $\sigma \in \mathfrak{S}_n$ and P is an ordered tree with n vertices whose non-root vertices are labeled by $[n - 1]$. An ordered tree, also called a plane tree, is a rooted tree for which siblings are given a linear order.
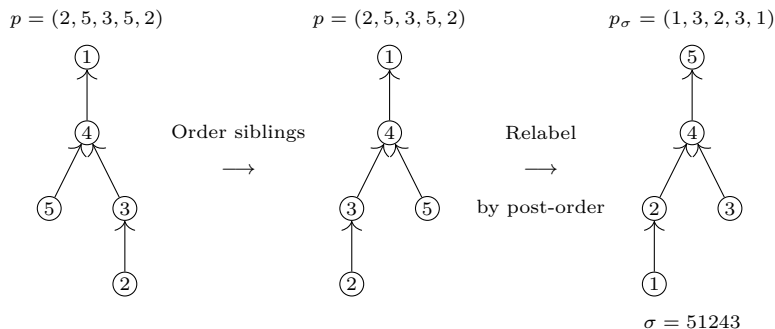
$p = (2, 5, 3, 5, 2)$ $p = (2, 5, 3, 5, 2)$ $p_\sigma = (1, 3, 2, 3, 1)$

Order siblings

$\longrightarrow$

Relabel

$\longrightarrow$

by post-order

$\sigma = 51243$

Fig. 6. An example of ϕ.

To describe ψ, we first give a definition. Recall post-order labeling: given an ordered tree $\mathcal{T}$, travel around the left border of the tree starting from the root, labeling in increasing order as one reaches the right side of a vertex. See the final picture in Fig. 6 for a tree labeled by post-order.

Definition 6. For an ordered tree $\mathcal{T}$ with $|\mathcal{T}| = n$ and sequence $p \in [n]^n$, we say the pair $(\mathcal{T}, p)$ is a *standardized restricted prime parking function* if

1. $(\mathcal{T}, p)$ is a prime parking function when $\mathcal{T}$ is considered an unordered tree.
2. For any pair of sibling vertices, $\{u, v\} \subseteq V(\mathcal{T})$, the vertex v is ordered to the right of u if and only if the edge (v, w), where w is the parent vertex, is crossed by some driver before the edge (u, w) is crossed during the parking procedure.
3. $\mathcal{T}$ is labeled via post-order.

We denote the set of standardized restricted prime parking functions on n vertices by $\mathcal{SRP}_n$. If $\mathcal{T}$ is an ordered tree, we say $(\mathcal{T}, p)$ is a prime parking function when the pair is a parking function if the ordering on $\mathcal{T}$ is forgotten.

The bijection ψ is constructed from two bijections, $\phi : (T, p) \mapsto (\sigma, (T_\sigma, p_\sigma))$ where $(T_\sigma, p_\sigma) \in \mathcal{SRP}_n$, and $\alpha : (T_\sigma, p_\sigma) \mapsto P$, where P is the aforementioned ordered tree with non-root vertices labeled by $[n-1]$. The unlabeled ordered trees on n vertices are counted by the Catalan number C_{n-1}, so including the labellings there are $C_{n-1}(n-1)!$-many such P. Combining these two steps, we conclude that $P_n = n!C_{n-1}(n-1)! = (2n-2)!$.

4.1. The bijection ϕ

We prove

Proposition 3. *For $n \geq 1$, we have*

$$P_n = n!|\mathcal{SRP}_n|.$$

Proof. Let $(T, p) \in \mathcal{PF}_n$. We induce an ordering on T by using p. Since the parking function is prime, every edge must be crossed by some driver after failing to park at her preferred spot. Additionally, since there is only one path a driver may travel in search of a spot and because drivers must park at the first available spot they encounter, the order in which edges are initially crossed is well-defined. Then, consider T as an ordered tree by ordering vertices to match property 2 in Definition 6.

For a $\sigma \in \mathfrak{S}_n$, let T_σ be the tree obtained by relabeling the vertices of T, $v \mapsto \sigma(v)$. Likewise, let $p_\sigma = (\sigma(p_1), \sigma(p_2), \ldots, \sigma(p_n))$. Then we define

$$\phi((T, p)) = (\sigma, (T_\sigma, p_\sigma)),$$

where σ is the unique permutation such that T_σ is labeled by post-order. This means $(T_\sigma, p_\sigma) \in \mathcal{SRP}_n$.

Since any two relabellings of an ordered tree are distinct, for $(\widetilde{T}, \widetilde{p}) \in \mathcal{SRP}_n$, the preimage $\phi^{-1}((\sigma, (\widetilde{T}, \widetilde{p}))$ is the prime parking function $(\widetilde{T}_{\sigma^{-1}}, \widetilde{p}_{\sigma^{-1}})$, where the ordering on the tree is forgotten. $\square$

Fig. 6 shows an application of ϕ. In the left tree, the edge $(5, 4)$ is first used by the fourth driver, while the edge $(3, 4)$ is not used until the fifth driver. Thus, the vertex 5 is placed to the right of vertex 3, which gives the ordered tree in the middle of the figure. The right tree is obtained from the middle one by relabeling via post-order, which gives the permutation σ. We now turn our attention to $|\mathcal{SRP}_n|$.

4.2. The bijection α

We define $\alpha : \mathcal{SRP}_n \to \{\text{ordered trees with non-root vertices labeled by } [n-1]\}$. The main observation necessary for the construction of α is a decomposition of $(\mathcal{T}, p) \in \mathcal{SRP}_n$ into an ordered collection of components, each a relabeling of a standardized restricted prime parking function, based on the edges used by the first $n - 1$ drivers. We use $\mathcal{T}$ to emphasize that $\mathcal{T}$ is an ordered tree. First, we give a proposition about where the final driver must park for prime parking functions, which also applies to parking functions in $\mathcal{SRP}_n$.

Proposition 4. *Let $(T, p) \in \mathcal{PF}_n$. Then the final driver parks at the root node.*

Proof. Let ω be the vertex the final driver parks at. If ω is not the root, it has a parent vertex v. Since drivers must park at the first empty vertex they arrive at, the edge (ω, v) can not be used by any driver prior to the final one, since ω is unoccupied. Since the final driver also does not use (ω, v), it remains unused. However, p is a prime parking function on T, so all edges must be used. Thus, there can be no edge (ω, v), and so ω is the root of T. $\square$
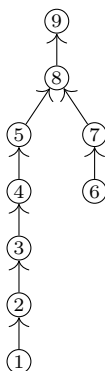
$$p = (6, 4, 1, 3, 3, 1, 6, 7, 2)$$

Fig. 7. A $(\mathcal{T}, p) \in \mathcal{SRP}_n$.

We first describe the recursive construction of α, then prove it is a bijection. Begin with $(\mathcal{T}, p) \in \mathcal{SRP}_n$. We use the tree in Fig. 7 as a running example.

Base Case. If $\mathcal{T}$ is a singleton, then $\alpha((\mathcal{T}, p))$ is an unlabeled singleton as $|\mathcal{SRP}_1| = 1$.

Step 1. Park all except the final driver, highlighting edges as they are used. Delete the non-highlighted edges and the root, marking the terminal vertex of any edge deleted and the vertex with label p_n.

Since p is a prime parking function on $\mathcal{T}$, the non-highlighted edges must lie on the path $\mathcal{P}$ between the vertex labeled p_n and the root. The highlighted edges define some collection of subtrees $\{\mathcal{T}_i\}_{i=1}^r$, linearly ordered by the order in which they are a part of $\mathcal{P}$. Since the root is always isolated, as the final driver is the only one to cross the edge connected to the root, we may ignore it. Let $p^{(i)}$ be the subsequence of $(p_1, \ldots, p_{n-1})$ consisting of all p_j such that $p_j \in V(\mathcal{T}_i)$. By construction, $p^{(i)}$ is a prime parking function on $\mathcal{T}_i$ satisfying conditions 1 and 2 in Definition 6. See Fig. 8 for this step applied to our running example. The non-highlighted edges are dotted and the marked vertices are shaded.

Step 2. For each $(\mathcal{T}_i, p^{(i)})$, let $A_i = \{j \in [n-1] : p_j \in V(\mathcal{T}_i)\}$. For $1 \leq i \leq r$, if the marked vertex on $\mathcal{T}_i$ has the kth smallest label among vertices in $\mathcal{T}_i$, mark the kth smallest element in A_i.

Notice that the elements of A_i are precisely those j such that p_j appears in $p^{(i)}$. The marked vertices and elements track how the $\mathcal{T}_i$ are connected to each other in $\mathcal{T}$. Fig. 9 shows this step. The elements of A_i are represented by C_j instead of j to emphasize that j is an index from p, rather than a vertex of $\mathcal{T}$.

Step 3. Relabel each $(\mathcal{T}_i, p^{(i)})$ so that its labels are in post-order, meaning $(\overline{\mathcal{T}}_i, \overline{p}^{(i)}) \in \mathcal{SRP}_n$. Apply α to each $(\overline{\mathcal{T}}_i, \overline{p}^{(i)})$.

In Fig. 10, we show components after they have been relabeled and apply α to each.

Step 4. For each ordered tree $\alpha((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))$, label the root with the marked element of A_i and relabel the rest of the vertices with the unmarked elements of A_i, preserving relative ordering. Denote these trees by $\{P_i\}_{i=1}^r$. Attach their roots to an unlabeled
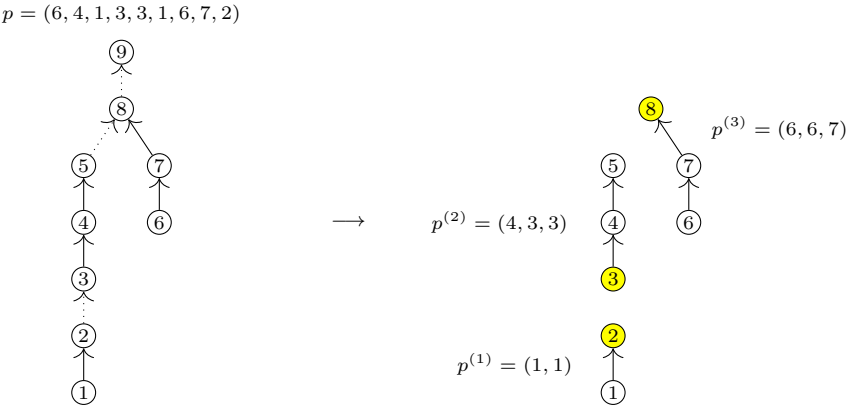
$p = (6, 4, 1, 3, 3, 1, 6, 7, 2)$

Fig. 8. Step 1, decomposing.

$p = (6, 4, 1, 3, 3, 1, 6, 7, 2)$

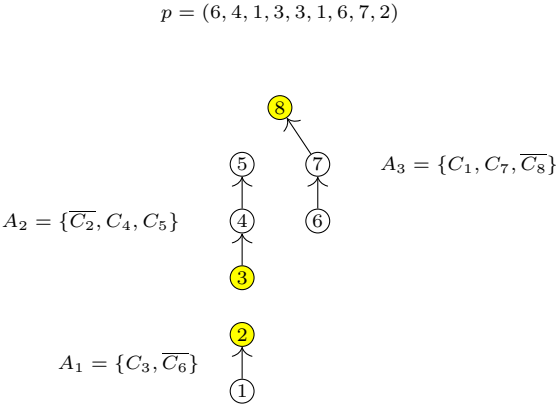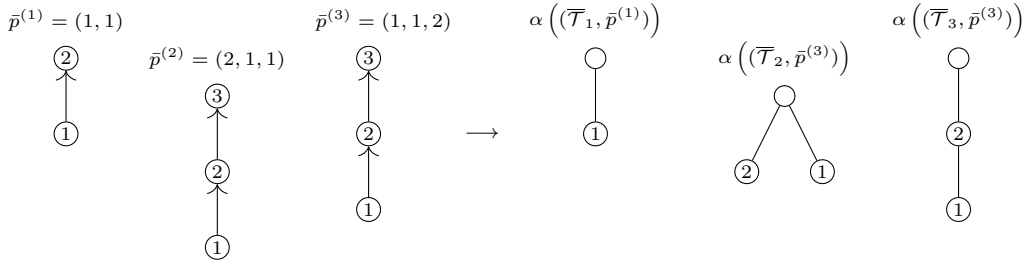**Fig. 9.** Step 2, tracking p and the shape of $\mathcal{T}$.

Fig. 10. Step 3, applying α on the components.

vertex and arrange the subtrees so that the subtree P_i is to the left of P_j if $i < j$. This is $\alpha((\mathcal{T}, p))$.

Fig. 11 shows the relabeling and Fig. 12 shows the final result of the running example. We constructed α to prove the following lemma.
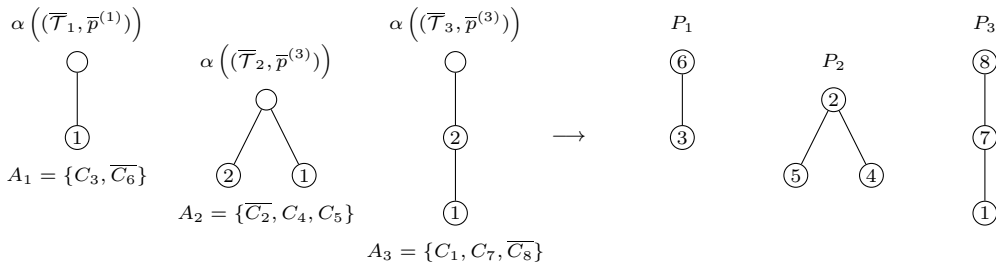
$\alpha\left((\overline{\mathcal{T}}_1, \overline{p}^{(1)})\right)$

$\alpha\left((\overline{\mathcal{T}}_3, \overline{p}^{(3)})\right)$

P_1

P_3

$\alpha\left((\overline{\mathcal{T}}_2, \overline{p}^{(3)})\right)$

$A_1 = \{C_3, \overline{C_6}\}$

$A_2 = \{\overline{C_2}, C_4, C_5\}$

$A_3 = \{C_1, C_7, \overline{C_8}\}$

Fig. 11. Step 4, relabeling the trees from Fig. 10 with the sets from Fig. 9.

$p = (6, 4, 1, 3, 3, 1, 6, 7, 2)$
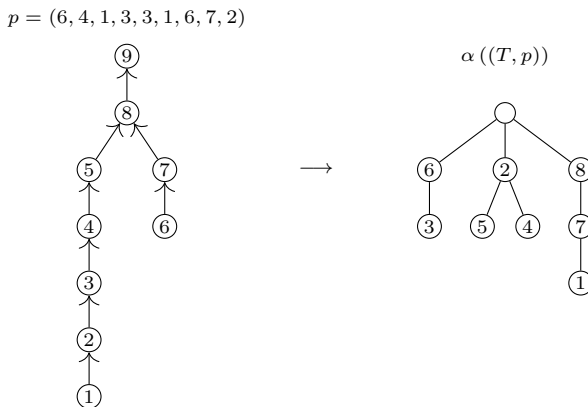
$\alpha\left((T, p)\right)$

Fig. 12. The result of α.

Lemma 1. *For $n \geq 1$*

$$|\mathcal{SPF}_n| = C_{n-1}(n-1)!$$

Proof. There are $C_{n-1}(n-1)!$ ordered trees with n vertices with non-root vertices labeled by $[n-1]$. Therefore, it is sufficient to show that α is a bijection, which we do inductively. In the case that $|T| = 1$, the only pair is $(T, 1)$ and $\alpha\left((T, 1)\right)$ is an unlabeled singleton.

Now suppose α is a bijection for all parking functions in $\mathcal{SRP}_k$ with $k < n$ and let $(\mathcal{T}, p) \in \mathcal{SRP}_n$. Run the parking procedure on T for all but the final driver. Since p is a prime parking function, all edges except for some on the path $\mathcal{P}$ from p_n to the root have been used. Deleting these unused edges creates a collection of trees which are naturally ordered by the order their vertices appear on $\mathcal{P}$. Since we know the root component is a singleton by Proposition 4, we may ignore it and label the other components $\{\mathcal{T}_i\}_{i=1}^r$ for some r.

We claim that $\mathcal{T}_i$ has smallest vertex label $1 + \sum_{j=1}^{i-1} |\mathcal{T}_j|$ and largest label $|\mathcal{T}_i| + \sum_{j=1}^{i-1} |\mathcal{T}_j|$. Further, if $\sum_{j=1}^{i-1} |\mathcal{T}_j|$ is subtracted from each vertex, the resulting tree will be labeled by post-order. This means if we consider the subsequence of p, denoted $p^{(i)}$,

consisting of drivers preferring $\mathcal{T}_i$, the pair $(\mathcal{T}_i, p^{(i)})$ is a relabeling of a parking function in $\mathcal{SRP}_{|\mathcal{T}_i|}$.

Proof of claim. Because the edge leaving the root of every $\mathcal{T}_i$ is not used until the very last driver and $(\mathcal{T}, p) \in \mathcal{SRP}_n$, the roots of the $\mathcal{T}_i$'s, denoted $\{\rho_i\}_{i=1}^r$, must lie on the left border of the tree.

By post-order labeling, a vertex is not labeled before all vertices below and all of its left siblings are given a label. Since all vertices below ρ_1 belong to $\mathcal{T}_1$, all other vertices of $\mathcal{T}_1$ are labeled before ρ_1. No vertex in $\mathcal{T}_2$ is labeled before ρ_1 since ρ_1 is on the left border of the tree, to the left of any of its siblings. Hence, $\mathcal{T}_1$ is labeled first and is in post-order.

In general, as ρ_{i-1} is on the left border of $\mathcal{T}$, the vertices below ρ_{i-1} are labeled before any vertex in $\mathcal{T}_i$. Thus, $\mathcal{T}_i$ has vertices labeled $1 + \sum_{j=1}^{i-1} |\mathcal{T}_j|$ to $|\mathcal{T}_i| + \sum_{j=1}^{i-1} |\mathcal{T}_j|$. Subtracting $\sum_{j=1}^{i-1} |\mathcal{T}_j|$ from each vertex is the same as labeling, via post-order, the maximal subtree with root ρ_i and vertices below ρ_{i-1} (inclusive) deleted, as this deletion removes a branch on the left side of the tree that is labeled before any other vertex. $\square$

Let $A_1, \ldots, A_r$ be a partition of $[n-1]$ such that $j \in A_\ell$ if and only if p_j is a vertex in $\mathcal{T}_\ell$. A_i is the set of indices of the drivers preferring the component $\mathcal{T}_i$, meaning $|A_i| = |\mathcal{T}_i|$. For each deleted edge (u, v), except for when v is the root, if v is the kth smallest vertex in its component, $\mathcal{T}_j$, mark the kth smallest element in A_j. For A_1, let $v = p_n$. These marked elements track both the final driver's preference and how to reconstruct the tree from its components. The collection $\{A_i\}_{i=1}^r$ partitions $[n-1]$ and will become the set of labels on the resulting ordered tree.

Relabel $\{(\mathcal{T}_i, p^{(i)})\}_{i=1}^r$ by post-order (notice this is the same as subtracting $\sum_{j=1}^{i-1} |\mathcal{T}_j|$ from $v \in V(\mathcal{T}_i)$), denoting them $\{(\overline{\mathcal{T}}_i, \overline{p}^{(i)})\}_{i=1}^r$. Use the inductive hypothesis to obtain the trees $\{\alpha((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))\}_{i=1}^r$. Label the root of $\alpha((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))$ with the marked vertex of A_i, then relabel the remaining vertices with the unmarked elements of A_i, preserving relative order. This is possible because $|A_i| = |\mathcal{T}_i| = |\alpha((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))|$. Attach the roots of these trees to an unmarked vertex and order them so that the tree using the labels in A_i is ith-from-the-left. This tree is $\alpha((\mathcal{T}, p))$.

To reverse, let P be an ordered tree with non-root vertices labeled by $[n-1]$. Deleting the root yields several components, denoted left-to-right as P_i for $1 \leq i \leq r$. The set A_i is the set of labels of P_i where the root of P_i is the marked element. Let k_i denote the relative size of the marked element in A_i. For each, delete the root's label and relabel using $[|P_i| - 1]$, preserving relative order, and apply α^{-1} to get the collection $\{(\overline{\mathcal{T}}_i, \overline{p}^{(i)})\}_{i=1}^r$. To each $v \in \overline{\mathcal{T}}_i$ and letter of $\overline{p}^{(i)}$, add $\sum_{j=1}^{i-1} |\overline{\mathcal{T}}_j|$ to recover the pairs $(\mathcal{T}_i, p^{(i)})$.

Then for $1 \leq i \leq r-1$, attach the root of $\mathcal{T}_i$ to the vertex labeled $k_i + \sum_{j=1}^{i-1} |\mathcal{T}_j|$, placing it to the left of any siblings. These vertices are those marked in Step 1 of the description of α. Attach the root of $\mathcal{T}_r$ to a singleton with label n, the root of $\mathcal{T}$. Let

the sequence $\{i_j\}$ be the increasing sequence of the elements of A_i. Then let $p_{i_j} = p_j^{(i)}$ to recover p. $\square$

Finally, we combine results to prove Theorem 1.

Proof of Theorem 1. Let (T, p) be a prime parking function, if $\phi((T, p)) = (\sigma, T_\sigma, p_\sigma)$, we define

$$\psi(T, p) = (\sigma, \alpha((T_\sigma, p_\sigma))).$$

The function ψ is a bijection by Proposition 3 and Lemma 1, so we conclude that

$$P_n = n!|\mathcal{SRP}_n| = n! \cdot (n-1)! \cdot C_{n-1} = (2n-2)!. \square$$

Remark 1. One of the anonymous referees noted that we can obtain this result via generating functions without the use of Equation (1). We may decompose an arbitrary prime parking function (T, p) by considering the tree right before the final driver parks. As in Step 1 of the description of α, this gives a sequence of prime parking functions, each with a marked leaf. Thus, accounting for the n labels on the tree and the first $n-1$ letters in p, we can say

$$P_n = \sum_{r \geq 0} \sum_{\substack{\sum_{i=1}^{r} k_i = n-1}} \binom{n}{k_1, \ldots, k_r, 1} \binom{n-1}{k_1, \ldots, k_r} \prod_{i=1}^{r} (P_{k_i} \cdot k_i).$$

Multiplying by $x^n/(n!)^2$, summing over $n \geq 1$, taking the derivative, and swapping the order of the summands, we see

$$P'(x) = \sum_{r \geq 0} (xP'(x))^r.$$

If we set $Q(x)$ to be the ordinary generating function for ordered trees, we know

$$Q(x) = \frac{x}{1 - xQ(x)},$$

so it follows that $xP'(x) = Q(x)$. However, we also know that $Q(x) = xC(x)$, so $P'(x) = C(x)$ and thus $P_n = (2n-2)!$.

5. Lucky drivers

For classical parking functions, Gessel and Seo [5] considered the number of drivers who successfully parked in their preferred spots. Such drivers were called lucky and motivate the following generalization.

Definition 7. Given a parking function (T, s), we say driver i is *lucky* if she parks at s_i. In other words, a lucky driver is one that parks at her preferred parking spot.

For a parking function (T, s), let $L((T, s))$ be the number of lucky drivers. Define the enumerator

$$\text{PTL}_n(t) = \sum_{(T,p) \in \mathcal{PF}_n} t^{L((T,p))}.$$

We prove Theorem 2 in order to immediately conclude

Corollary 2. $PTL_1(t) = t$ and for $n \geq 2$, we have

$$PTL_n(t) = n!(n-1)! \sum_{k=1}^{n-1} N_{n-1,k} t^k$$

$$= n!(n-2)! \sum_{k=1}^{n-1} \binom{n-1}{k} \binom{n-1}{k-1} t^k.$$

Proof of Theorem 2. For $n \geq 1$, let $(T, p) \in \mathcal{PF}_n$. We claim $L((T, p))$ is the number of leaves in the tree $\alpha((T_\sigma, p_\sigma))$, the second component of $\psi(T, p)$. When $n = 1$, there is precisely one prime parking function and one lucky driver and the only tree under α is a singleton, which has one leaf.

For $n \geq 2$, suppose the claim is true for $m < n$. Recall from Proposition 4 that the final driver must park at the root, but can not prefer the root, and thus is not lucky. Therefore, $L((T, p)) = \sum_{i=1}^{r} L((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))$ where $\{(\overline{\mathcal{T}}_i, \overline{p}^{(i)})\}_{i=1}^{r}$ are the standardized parking functions from Step 3 in the description of α. Since $|\overline{\mathcal{T}}_i| < n$, the inductive hypothesis tells us that $L((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))$ is the number of leaves in $\alpha((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))$. The shape of $\alpha((T_\sigma, p_\sigma))$ is obtained by attaching the roots of the trees $\{\alpha((\overline{\mathcal{T}}_i, \overline{p}^{(i)}))\}_{i=1}^{r}$ to a new root, so $L((T, p))$ is indeed the number of leaves in $\alpha((T_\sigma, p_\sigma))$.

By Lemma 1, there are $(n-1)!$-many choices of parking functions (T', p') such that $\alpha((T', p'))$ has the same shape as $\alpha((T_\sigma, p_\sigma))$. Proposition 3 tells us there are $n!$-many choices of parking functions whose representative in $\mathcal{SRP}_n$ is (T_σ, p_σ). Finally, the Narayana number $N_{n-1,k}$ counts the number of ordered plane trees with n vertices and k leaves. $\quad\square$

Using the parking function and corresponding plane tree in Fig. 12, we note that $\alpha((T, p))$ has 4 leaves while the first four drivers are the four lucky drivers.

6. Preimages of paths under α

We investigate special families of trees and parking functions of interest.

6.1. Preimage of paths

We study what kinds of parking functions (T, p) appear under α^{-1} when we restrict the domain to trees which are paths. Such paths will have $n+1$ nodes with the non-root nodes labeled by $[n]$.

Definition 8. We denote the paths in the domain of α^{-1} by P_σ, where $\sigma \in \mathfrak{S}_n$ is the permutation obtained by reading labels traveling away from the root.

For a tree of size $n+1$, there are $n!$ such paths. Recall that $\mathcal{P}_n$ is the path of n vertices upon which classical parking functions are defined.

Proposition 5. *Let $(\mathcal{P}_{n+1}, s)$ be a parking function satisfying $s_1 = 1$ and $s_i \le i - 1$ for $i \ge 2$. Then $\alpha(\mathcal{P}_{n+1}, s)$ is one of the $n!$ paths with non-root vertices labeled by $[n]$.*

Proof. That s is prime is easily checked. Since $s_1 = 1$, we may delete s_1 and consider $s' = (s_2, s_3, \ldots, s_{n+1})$. Since $s'_i = s_{i+1} \le i$, we have $|\{i : s'_i \le k\}| \ge k$ for any $k \in [n]$, so s' is a parking function on $\mathcal{P}_n$. Further, s'_i may be one of i-many choices, so there are $n!$-many s', and thus s.

Park the drivers in order. The first driver takes spot 1 and since $s_2 = 1$, the second driver takes spot 2, crossing the edge from 1 to 2. Next, since $s_2 \le 2$, and spots 1 and 2 are taken, the third driver takes spot 3, crossing the edge from 2 to 3. Continuing this, the ith driver always parks at, but never prefers, spot i. When the final driver parks, all spots except for $n + 1$ are filled and every edge has been used except the one between n and $n + 1$. Thus, by the construction of α, the root of $\alpha(s, \mathcal{P}_{n+1})$ has one child. The inductive step in the proof of Lemma 1 tells us that the shape of the tree obtained by deleting the root from $\alpha(s, \mathcal{P}_{n+1})$ is the same as that of $\alpha\left((1, s_1, \ldots, s_{n-1}), \mathcal{P}_n\right)$. But $(1, s_1, \ldots, s_{n-1})$ is a prime parking function on $\mathcal{P}_n$ with the same growth property as s. Therefore, its root also has one child. Iterating this argument, we see that the image of such parking functions under α consists of paths with non-root vertices labeled by $[n]$. Since α is a bijection and there are $n!$ choices for s, all $n!$-such paths must appear. $\square$

It will prove useful to have a characterization of the s_i in terms of the labels of $\alpha(\mathcal{P}_{n+1}, s)$.

Lemma 2. *Let $(\mathcal{P}_{n+1}, s)$ be a prime parking function such that $s_1 = 1$ and $s_i \le i - 1$ for $2 \le i \le n + 1$ and let $P_\sigma = \alpha(\mathcal{P}_{n+1}, s)$ for $\sigma \in \mathfrak{S}_n$. Then for $2 \le i \le n + 1$, we have*

$$s_i = |\{j > n + 2 - i : \sigma_j < \sigma_{n+2-i}\}| + 1.$$

Proof. By the construction of α in Lemma 1, we know $\sigma_1 = s_{n+1}$ and so σ_2 is the element in $[n] \setminus \{\sigma_1\}$ larger than exactly $s_n - 1$ others (the s_nth smallest element). In general, σ_i is the s_{n+2-i}th smallest element in $[n] \setminus \{\sigma_1, \sigma_2, \ldots, \sigma_{i-1}\}$. Thus, for $i \ge 2$, s_i is given

$p = (1, 1, 2, 3, 3, 1)$ $\alpha\left((\mathcal{P}_6, p)\right) = \mathcal{P}_\sigma$ $p = (1, 1, 2, 3)$ $\alpha\left((\mathcal{P}_4, (1, 1, 2, 3))\right)$
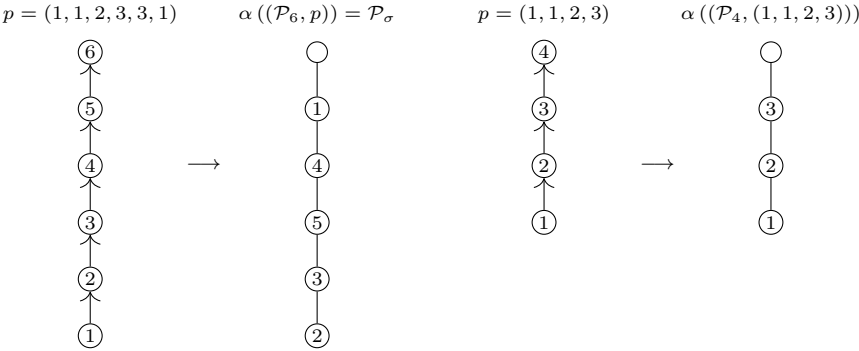

Fig. 13. Applying α to several classical parking functions.

by the relative size of σ_{n+2-i} in the set $[n] \setminus \{\sigma_1, \sigma_2, \ldots, \sigma_{n+1-i}\}$. We may write this as $s_i = |\{j > n + 2 - i : \sigma_j < \sigma_{n+2-i}\}| + 1$. $\quad \square$

Example 1. Consider the classical prime parking function $p = (1, 1, 2, 3, 3, 1)$ and tree $\mathcal{P}_6$. That $s_i \le i - 1$ for $i \ge 2$ means the root of $\alpha((\mathcal{P}_6, p))$ has only one child. Further, as the final driver has preference 1, the tree $\alpha((\mathcal{P}_6, p))$ is constructed by labeling the root of $\alpha((\mathcal{P}_5, (1, 1, 2, 3, 3))$ with 1 and re-labeling the other nodes to preserve relative order.

To determine σ_2, we note that the root child of $\alpha((\mathcal{P}_5, (1, 1, 2, 3, 3)))$ is 3 because the final driver prefers 3. But for $\alpha((\mathcal{P}_6, p))$, we must adjust to account for the 1 used and in order to preserve relative order. Therefore, σ_2 is the 3rd smallest in $[6] \setminus \{1\}$, which is 4.

Continuing this trend, σ_3 is the label of the root child of $\alpha((\mathcal{P}_4, (1, 1, 2, 3)))$ adjusted to account for the labels 1 and 4 already having been used. The final driver prefers 3, so σ_3 is the third smallest in the set $[6] \setminus \{1, 4\}$, which is 5. Determining σ_i for $i \ge 4$ proceeds in a similar fashion.

Fig. 13 presents $\alpha((\mathcal{P}_6, p))$ and $\alpha((\mathcal{P}_4, (1, 1, 2, 3)))$. Note that the leaf of $\alpha((\mathcal{P}_6, p))$ and its two generations of parents have the same relative order as the labeled vertices in $\alpha((\mathcal{P}_4, (1, 1, 2, 3)))$.

Of particular interest are the increasing parking functions that obey this growth restriction. We now examine which labellings σ appear for $\alpha((\mathcal{P}_{n+1}, s))$ with s an increasing prime parking function.

6.2. Image of classical increasing parking functions

Borie [2] gives a bijection between $\mathfrak{S}_n(132)$, the permutations of length n which avoid a 132 pattern, and classical increasing parking functions of length n. Let $\sigma \in \mathfrak{S}_n(132)$. Define for $m \in [n]$:

$$f_\sigma(m) = |\{i : |\{j : j < i, \sigma_j > \sigma_i\}| \ge m\}|.$$

We briefly note that our $f_\sigma(m)$ is equivalent to the function $mmp(0, m, 0, 0)(\sigma)$ in [2], which we have shortened for readability. Set

$$\phi(\sigma) = (f_\sigma(n) + 1, f_\sigma(n-1) + 1, \ldots, f_\sigma(1) + 1).$$

Then $\phi(\sigma)$ is a classical increasing parking function and we have the following theorem due to Borie:

Theorem 4 *(Theorem 3.3, [2]). ϕ is a bijection between $\mathfrak{S}_n(132)$ and increasing parking functions of length n.*

We will show

Theorem 5. *For $\sigma \in \mathfrak{S}_n(132)$, $\phi(\sigma)$ is the parking function obtained after deleting the leading 1 from $\alpha^{-1}(P_\sigma)$.*

Proof. Fix $\sigma \in \mathfrak{S}_n(132)$ and let $(\mathcal{P}_{n+1}, p') = \alpha^{-1}(P_\sigma)$. By Proposition 5, we know we may write $p' = (1, p_1, p_2, \ldots, p_n)$, where $p = (p_1, p_2, \ldots, p_n)$ is a classical increasing parking function. For brevity, define for $m \in [n]$:

$$A_m = \{j : j > m \text{ and } \sigma_j < \sigma_m\},$$

and

$$B_m = \{i : |\{j : j < i, \sigma_j > \sigma_i\}| \geq m\}$$
$$= \{i > m : \sigma_i \text{ smaller than at least } m \text{ of } \{\sigma_1, \ldots, \sigma_{i-1}\}\}.$$

Since $|B_m| = f_\sigma(m)$, we have $\phi(\sigma) = (|B_n| + 1, |B_{n-1}| + 1, \ldots, |B_1| + 1)$. From Lemma 2 we have $p_i = p'_{i+1} = |A_{n+1-i}| + 1$, so it is sufficient to show that $|A_m| = |B_m|$ for $m \in [n]$. We show the sets are the same.

Fix $m \in [n]$ and let $k \in A_m$. By definition, $m < k$ and $\sigma_k < \sigma_m$. For $i < m$, if $\sigma_i < \sigma_k$, then σ has the 132 pattern $\sigma_i \sigma_m \sigma_k$, which is not possible. Hence, $\sigma_k < \sigma_i$ for $1 \leq i \leq m$, so $k \in B_m$.

On the other hand, let $j \in B_m$. If $\sigma_j < \sigma_m$, then $j \in A_m$, so suppose $\sigma_j > \sigma_m$. Since $\sigma \in \mathfrak{S}_n(132)$, $\sigma_i < \sigma_j$ for $m+1 \leq i \leq j-1$. Thus, for indices smaller than j, only elements from $\{\sigma_1, \sigma_2, \ldots, \sigma_{m-1}\}$ may be larger than σ_j. However, $|\{\sigma_1, \sigma_2, \ldots, \sigma_{m-1}\}| = m-1 < m$, contradicting that $j \in B_m$. Therefore $\sigma_j < \sigma_m$, so $j \in A$. □

7. Parking distributions

Recall that a parking distribution is a parking function (T, s) such that the entries of s are weakly increasing. Following the decomposition in Section 3, we let $\widetilde{F}_n$ be the total

number of parking distributions on trees with n vertices, and $\widetilde{P}_n$ be the corresponding number of prime parking distributions. Set

$$\widetilde{F}(x) = \sum_{n \geq 1} \widetilde{F}_n \frac{x^n}{n!} \qquad \text{and} \qquad \widetilde{P}(x) = \sum_{n \geq 1} \widetilde{P}_n \frac{x^n}{n!}.$$

Notice that these are exponential generating functions, unlike $F(x)$ and $P(x)$, as we no longer need to account for the order of the preference sequences. Decomposing a parking distribution (T, s) into the "core" prime component and collection of r other components, as in Section 3, we get

$$\widetilde{F}_n = \sum_{r \geq 0} \frac{1}{r!} \sum_{\substack{\sum\limits_{i=0}^{r} k_i = n}} \widetilde{P}_{k_0} \widetilde{F}_{k_1} \cdots \widetilde{F}_{k_r} \binom{n}{k_0, k_1, \ldots, k_r} (k_0)^r.$$

Summing over n, we find

$$\widetilde{F}(x) = \widetilde{P}\left(x e^{\widetilde{F}(x)}\right), \tag{5}$$

which is the same relationship for parking functions in Equation (2). We then turn our attention to prime parking distributions and prove

Theorem 3. *The total number of prime parking distributions on trees with $n \geq 1$ vertices is given by*

$$\widetilde{P}_n = (n-1)! S_{n-1},$$

where $\{S_i\}_{i \geq 0}$ are the large Schröder numbers.

Proof. We let P_n^* be the total number of tuples (T, p, v), called *marked* prime parking distributions, where (T, p) is a prime parking distribution and v is a leaf of T with $|T| = n$. Define the exponential generating function

$$P^*(x) = \sum_{n \geq 1} P_n^* \frac{x^n}{n!}.$$

We count P_n^* in two ways in order to determine the coefficients of $\widetilde{P}(x)$. We may construct a marked prime parking distribution from a prime parking distribution (T, p) with $|T| = n - 1$ by "growing" the marked leaf from some vertex in T. We choose one vertex w in T which has $j \geq 1$ drivers preferring it, select a label for the marked leaf v (out of n choices), add 1 to the label of any vertex with label greater than or equal to the label chosen for v, attach v as a child, add a driver preferring v, and change $1 \leq i \leq j$ drivers
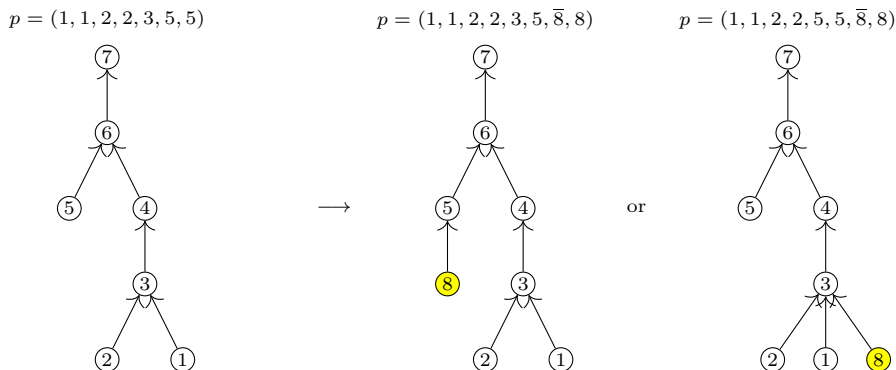
$p = (1, 1, 2, 2, 3, 5, 5)$ $p = (1, 1, 2, 2, 3, 5, \overline{8}, 8)$ $p = (1, 1, 2, 2, 5, 5, \overline{8}, 8)$

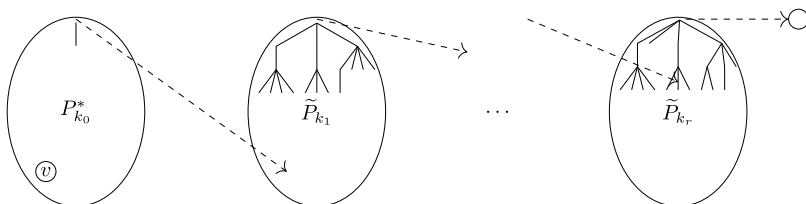**Fig. 14.** Two possibilities when adding the marked leaf.

Fig. 15. Decomposition based on final driver's movement.

preferring w to prefer v instead. The number of choices we can make for v's parents and number of drivers whose preferences we change is $\sum_{w \in V(T)} |\{i : p_i = w\}| = n - 1$.

On the other hand, any marked prime parking distribution can be changed to a regular prime parking distribution by deleting the marked vertex v, deleting one driver preferring it, and changing the preference of other drivers preferring v to instead prefer v's parent. Since the marked parking distribution is prime, at least two drivers prefer the marked leaf. Fig. 14 has two examples where the shaded vertex is added and marked letters of p are drivers whose preference was changed.

This means for $n \geq 2$,

$$P_n^* = n(n-1)\widetilde{P}_{n-1},$$

so noting $P_0^* = 0$ and $P_1^* = 1$ and summing over n, we get

$$P^*(x) = x + x^2 \widetilde{P}'(x). \tag{6}$$

For the second equation, we decompose a marked prime parking distribution (T, p, v) as in Section 4.2: park all drivers except for one preferring v. For a general picture, see Fig. 15. Dashed edges denote those unused before the final driver. As before, the edges which have not yet been used define $r + 1$, for some $r \geq 0$, components of size k_i with prime parking functions, one of which is marked. Accounting for the edges connecting

the components, the label of the root, and the labels on the components, we have for $n \geq 2$,

$$P_n^* = n \sum_{r \geq 0} \sum_{\sum k_i = n-1} \binom{n-1}{k_0, k_1, \ldots, k_r} P_{k_0}^* \widetilde{P}_{k_1} \cdots \widetilde{P}_{k_r} k_1 k_2 \cdots k_r,$$

so multiplying by $\frac{x^n}{n!}$ and summing over $n \geq 2$, we get

$$\sum_{n \geq 2} P_n^* \frac{x^n}{n!} = x \cdot \sum_{n \geq 2} \sum_{r \geq 0} \sum_{\sum k_i = n-1} \frac{P_{k_0}^*}{k_0!} \cdot \frac{\widetilde{P}_{k_1}}{(k_1 - 1)!} \cdots \frac{\widetilde{P}_{k_r}}{(k_r - 1)!} x^{n-1}$$

$$= x \cdot \sum_{n \geq 2} \sum_{k_0 = 1}^{n-1} \frac{P_{k_0}^*}{k_0!} \cdot [x^{n-1-k_0}] \frac{1}{1 - x\widetilde{P}'(x)} x^{n-1}.$$

After changing indices and noting $P_0^* = 0$ and $P_1^* = 1$, we see

$$P^*(x) = x + \frac{xP^*(x)}{1 - x\widetilde{P}'(x)}. \tag{7}$$

Combining Equations (6) and (7), we see

$$x\left(\widetilde{P}'(x)\right)^2 + (x - 1)\widetilde{P}'(x) + 1 = 0,$$

and so as $\widetilde{P}'(0) = 1$,

$$\widetilde{P}'(x) = \frac{1 - x - \sqrt{x^2 - 6x + 1}}{2x},$$

which is the ordinary generating function for the large Schröder numbers. Hence,

$$\widetilde{P}'(x) = \sum_{n \geq 0} n! S_n \frac{x^n}{n!},$$

meaning

$$\widetilde{P}(x) = \sum_{n \geq 1} (n-1)! S_{n-1} \frac{x^n}{n!}. \quad \square$$

In addition to the relationship in Equation (5), we can let F_n^* denote the total number of parking distributions on trees with n vertices with one leaf marked and let

$$F^*(x) = \sum_{n \geq 1} F_n^* \frac{x^n}{n!}.$$

Constructing a parking distribution counted by F_n^* by "growing" it from a parking distribution counted by $\widetilde{F}_{n-1}$ gives n choices for the leaf label, $n-1$ choices of nodes to attach the marked leaf to without reassigning drivers, and $n-1$ choices for attaching the marked leaf to a node and reassigning at least one driver from the parent node. Thus, $F_n^* = 2n(n-1)\widetilde{F}_{n-1}$, so

$$F^*(x) = x + 2x^2 \widetilde{F}'(x). \tag{8}$$

On the other hand, if a parking distribution on a marked tree with n vertices is decomposed by considering the final unfilled node when the drivers on the marked leaf park last, the final node is of one of three types: the root, the marked leaf, or neither. Therefore, we have for $n \geq 2$

$$F_n^* = \sum_{r \geq 0} \frac{n}{r!} \sum_{\sum_{i=0}^{r} k_i = n-1} \binom{n-1}{k_0, \ldots, k_r} F_{k_0}^* \widetilde{F}_{k_1} \cdots \widetilde{F}_{k_r}$$

$$+ n(n-1)F_{n-1}$$

$$+ \delta_{n \geq 3} \sum_{r \geq 0} \frac{n}{r!} \sum_{k + \sum_{i=0}^{r} k_i = n-1} \binom{n-1}{k_0, \ldots, k_r, k} F_{k_0}^* \widetilde{F}_{k_1} \cdots \widetilde{F}_{k_r} \widetilde{F}_k k,$$

where $\delta_{n \geq 3}$ is 1 if $n \geq 3$ and 0 otherwise. Summing over n we get

$$F^* = x + x F^* e^F + x^2 F' + x^2 F^* F' e^F. \tag{9}$$

Combining Equations (8) and (9) proves

Theorem 6. *The exponential generating function of parking distributions on trees, $\widetilde{F}(x)$, satisfies the differential equation*

$$\widetilde{F}' = e^{\widetilde{F}}(1 + x\widetilde{F}')(1 + 2x\widetilde{F}'),$$

with initial condition $\widetilde{F}(0) = 0$.

We note that the generating function for general tree parking functions, $F(x)$, satisfies the similar equation

$$F' = e^F (1 + xF')^2.$$

For details, see Equation (7) in [9].

8. Conclusion

This paper extends the results by Lackner and Panholzer [9] to prime parking functions on trees, from which all tree parking functions can be constructed. Additionally, we enumerate prime parking distributions, the natural generalization of increasing classical parking functions. There are many open questions regarding tree parking functions, of which we mention several:

1. Find an "interesting" direct bijection from prime tree parking functions to $\mathfrak{S}_{2n-2}$, preferably preserving some statistics.
2. Are there families of trees, such as caterpillars or binary trees, which support an interesting sequence of prime parking functions?
3. In [9], the authors find bounds for the number of parking functions, $S(T, n)$, on a given tree T and ask if $S(T, n)$ has a "simple characterization". We ask the same question for prime parking functions on a given tree.
4. Butler, Graham, and Yan [3] consider parking distributions on trees in the case where $n + k$ drivers attempt to park and exactly k do not find a space. Can their results be refined to consider only prime parking distributions?

Acknowledgments

References

[1] D. Armstrong, N.A. Loehr, G.S. Warrington, Rational parking functions and Catalan numbers, Ann. Comb. (ISSN 0219-3094) 20 (1) (Mar 2016) 21–58, https://doi.org/10.1007/s00026-015-0293-6.
[2] N. Borie, On the combinatorics of quadrant marked mesh patterns in 132-avoiding permutations, Australas. J. Combin. 64 (1) (2016) 140–153, https://hal.archives-ouvertes.fr/hal-01222921.
[3] S. Butler, R. Graham, C.H. Yan, Parking distributions on trees, European J. Combin. (ISSN 0195-6698) 65 (2017) 168–185, https://doi.org/10.1016/j.ejc.2017.06.003, http://www.sciencedirect.com/science/article/pii/S019566981730077X.
[4] R. Ehrenborg, A. Happ, Parking cars of different sizes, Amer. Math. Monthly 123 (10) (2016) 1045–1048, https://doi.org/10.4169/amer.math.monthly.123.10.1045.
[5] I.M. Gessel, S. Seo, A refinement of Cayley's formula for trees, Electron. J. Combin. 11 (2) (2006), http://www.combinatorics.org/Volume_11/Abstracts/v11i2r27.html.
[6] W. King, C.H. Yan, Parking functions on oriented trees, Sém. Lothar. Combin. 80B (2018) 47, https://www.mat.univie.ac.at/~slc/wpapers/FPSAC2018/47-King-Yan.pdf.
[7] A.G. Konheim, B. Weiss, An occupancy discipline and applications, SIAM J. Appl. Math. (ISSN 0036-1399) 14 (6) (1966) 1266–1274, http://www.jstor.org/stable/2946240.
[8] J.P. Kung, C. Yan, Gončarov polynomials and parking functions, J. Combin. Theory Ser. A (ISSN 0097-3165) 102 (1) (2003) 16–37, https://doi.org/10.1016/S0097-3165(03)00009-8, http://www.sciencedirect.com/science/article/pii/S0097316503000098.
[9] M.-L. Lackner, A. Panholzer, Parking functions for mappings, J. Combin. Theory Ser. A (ISSN 0097-3165) 142 (2016) 1–28, https://doi.org/10.1016/j.jcta.2016.03.001, http://www.sciencedirect.com/science/article/pii/S0097316516000236.

[10] A. Postnikov, B. Shapiro, Trees, parking functions, syzygies, and deformations of monomial ideals, Trans. Amer. Math. Soc. 142 (2004).

[11] R. Stanley, Enumerative Combinatorics, vol. 2, Cambridge University Press, ISBN 9780521560696, 1999, https://books.google.com/books?id=7-OXjgEACAAJ.

[12] C.H. Yan, Parking functions, in: M. Bóna (Ed.), Handbook of Enumerative Combinatorics, Chapter 13, CRC Press, 2015, pp. 835–894.