Mappings of Acyclic and Parking Functions

DOMINIQUE FOATA and JOHN RIORDAN
(Gainesville, Florida, U.S.A. and New York, New York, U.S.A.)

Abstract

The two functions in question are mappings: $[n]\to[n]$, with $[n]=\{1, 2,..., n\}$. The acyclic function may be represented by forests of labeled rooted trees, or by free trees with $n+1$ points; the parking functions are associated with the simplest ballot problem. The total number of each is $(n+1)^{n-1}$. The first of two mappings given is based on a simple mapping, due to H. O. Pollak, of parking functions on tree codes. In the second, each kind of function is mapped on permutations, arising naturally from characterizations of the functions. Several enumerations are given to indicate uses of the mappings.

1. Introduction

As is well known (see e.g. [7]) any of the n^n maps $f: [n]\to[n]$, $[n]=\{1, 2,..., n\}$, $n=1, 2,...$, may be represented by a linear graph with n labeled points as follows: a fixed point $f(x)=x$ is a sling at point x (which may be taken as identifying a root of the graph), $f(x)=y$, $y\neq x$, is a line from x to y. The graph has one or more connected components, each of which contains a single cycle of length (number of lines in the cycle) at most n, a sling counting as a cycle of length 1. For cycles of length at least three, the lines in cycle may be directed either clockwise or counterclockwise. The acyclic functions are maps represented by graphs with slings and no cycles of length greater than one; these graphs are labeled forests of rooted trees, the slings identifying the roots. Such forests are mapped on (free) trees with $n+1$ labeled points by the simple device (cf. [1], p. 85) of connecting all rooted points to a new point labeled $n+1$ (and removing the root indicators). Hence the number of acyclic functions is $(n+1)^{n-1}=T_{n+1}$, by Cayley's formula.

It is clear that each of the three: acyclic functions, labeled forests of rooted trees, and free trees, determines the other two; they will be used interchangeably hereafter.

The parking maps, $g: [n]\to[n]$, seem to have been introduced by A. G. Konheim and B. Weiss in [5] as a picturesque way of enlivening their work on computer storage problems. They described them as follows: there are n numbered parking spaces on a one-way street and n consecutive parkers seeking to fill them. Each parker enters with a parking preference, any number from 1 to n, say $g(i)$ for parker i. If space $p(i)$ is filled when parker i arrives, he continues to the next free space, if any. The parking functions: $\{g(1),..., g(n)\}$ are those preference sequences permitting all n parkers to park.

Received April 27, 1971 and in revised form January 3, 1972

```
  g        Code      tree        f

1 2 3      1 1       2 3       1 1 1
                     \ /
                      1
                      |
                      4

1 3 1      2 2       1 3       2 2 2
                     \ /
                      2
                      |
                      4

3 2 1      3 3       1 2       3 3 3
                     \ /
                      3
                      |
                      4

3 1 2      2 1     3 2 1 4     1 1 2
2 1 2      3 1     2 3 1 4     1 3 1
2 3 1      1 2     3 1 2 4     2 2 1
2 1 3      3 2     1 3 2 4     3 2 1
1 2 1      1 3     2 1 3 4     3 1 3
1 3 2      2 3     1 2 3 4     2 3 3
1 1 3      4 2     1 4 2 3     1 2 2
2 2 1      4 3     1 4 3 2     1 3 3
1 1 2      4 1     2 4 1 3     1 2 1
2 1 1      3 4     2 4 3 1     3 2 3
1 2 2      1 4     3 4 1 2     1 1 3
3 1 1      2 4     3 4 2 1     2 2 3
1 1 1      4 4      1 2 3      1 2 3
                    \ | /
                      4
```

Figure 1

As shown in [5], the numbers of parking functions is also $(n+1)^{n-1}$, which raises the question of mapping these two kinds of functions on each other.

One such mapping, by M. P. Schützenberger [10], has already appeared.

Here we present two different mappings, which seem to us both simple and natural. The first, which will be presented in the next section, rests on a mapping, by H. O. Pollak, of parking functions on tree codes (which has been announced in [9]). Our contribution is the proof that it is a bijection. The second is of completely different character. Though both functions are mapped on permutations, which seem to have the role of tree codes in the first mapping, these mappings are not bijections. Instead, the permutations appear in the mapping of the functions. If $\sigma_f(x)$ is the permutation associated with acyclic function f, $\tau_g(x)$ the similar permutation for parking function g, the second mapping may be put concisely as

$$\left.\begin{array}{ll} g(x)=1, & f(x)=x \\ g(x)=1+\sigma_f(f(x)), & f(x)\neq x \end{array}\right\} \qquad (1.1)$$

and

$$\left.\begin{array}{ll} f(x)=x, & g(x)=1 \\ f(x)=\tau_g^{-1}(g(x)-1), & g(x)\neq 1. \end{array}\right\} \qquad (1.2)$$

$\tau_g^{-1}(x)$ of course is the inverse of $\tau_g(x)$: $\tau_g^{-1}(x)\,\tau_g(x)=I$, the identity permutation.

The two permutations are defined in Section 3, but we may note here that $\tau_g(x)$ may be written succinctly as

$$\tau_g(x)=\mathrm{Card}\{y\in[n]: g(y)<g(x), \quad \text{or} \quad g(y)=g(x) \quad \text{and} \quad y\leqslant x\}.$$

Recently J. Françon [3] has given all the labelings of the elements of $[n]$ which make possible similar mappings. Surprisingly, each of these corresponds to a *selection procedure*, a familiar topic in computer programming circles.

Finally, the last section examines some consequences of the mappings, as an indication of their uses.

2. The First Mapping

Pollak's mapping of parking functions on tree codes is simplicity itself: the code for parking function $(g(1),\ldots,g(n))$ is $(c(1),\ldots,c(n-1))$, with

$$c(i)=g(i+1)-g(i) \bmod \overline{n+1}. \qquad (2.1)$$

Notice that the inverse of (2.1) may be written as

$$g(i)=g(1)+c(1)+\cdots+c(i-1), \bmod \overline{n+1}, \quad i=2(1)\,n; \qquad (2.2)$$

that is, a parking function is determined by its code and (2.2) only if $g(1)$ is known independently.

We now prove: (i) the numbers of codes is equal to the number of functions[1]) (T_{n+1}), and (ii) for any code there is a unique $g(1)$, determined by a property of parking functions.

---

[1]) We owe the germ of this proof to a referee.

For the first we repeat Pollak's proof (already given in [9]) that the number of parking functions is T_{n+1}, as follows. The parking preference sequences $\{p_1, \ldots, p_n\}$, $p_i = 1(1)n$, $i = 1(1)n$ are replaced by $\{p_1, \ldots, p_n\}$, $p_i = 1(1)\overline{n+1}$, $i = 1(1)n$; the one way street with n numbered spaces is replaced by a parking circle with $n+1$ spaces numbered clockwise (say). Preference $p_i = n+1$ is treated like any other preference: if space $n+1$ is occupied, parker i moves clockwise to the first empty space. For every preference sequence there is one empty space, and by symmetry there are exactly as many sequences with a given empty space as with any other. Since the total number of sequences is $(n+1)^n$, the number with a given empty space, in particular the number of parking functions whose empty space is $n+1$, is $(n+1)^{n-1}$. Moreover, the sequences with empty space i are the parking functions on the set $\{i+1, \ldots, n+1, 1, \ldots, i-1\}$, a cyclic renumbering. The coding is independent of this renumbering; each class of sequences, determined by a given empty space, has the same set of codes, the size of which must be $(n+1)^{n-1}$.

EXAMPLE. Take the simplest nontrivial case $n=2$; the classes of sequences may be arrayed as follows:

Empty space	Sets		
3	11	12	21
1	22	23	32
2	33	31	13

For each class the codes are (0, 1, 2), mod 3.

We turn now to the parking function property that serves to identify $g(1)$, given the code. First, any permutation of the elements of a parking function (which simply renumbers the parkers) is also a parking function. Next, following [9], the parking function elements may be regarded as partitions of some number (their sum) and the alternative partition representation $1^{r_1} 2^{r_2} \ldots n^{r_n}$, with $r_1 + \cdots + r_n = n$, is a *specification* independent of element permutations; r_i is the number of elements equal to i, that is: card $\{x, g(x) = i\}$. It is clear that $r_n \leqslant 1$, since if two or more parkers chose n, only one may park; a similar argument shows that $r_{n-j+1} + \cdots + r_n \leqslant j$, $j = 1(1)\overline{n-1}$. With $R_j = r_1 + \cdots + r_j - j$, these conditions are equivalent to

$$R_j \geqslant 0, \quad j = 1(1)\overline{n-1}, \quad R_n = 0. \tag{2.3}$$

It is shown in [9] that the number of parking function specifications is the ballot number $a_{nn} = a_{n,n-1}$, with

$$a_{nk} = \binom{n+k}{k} - \binom{n+k}{k-1}.$$

This implies that the ordered parking functions (monotonic non-decreasing) are in number a_{nn}, which is the number of weak lead lattice paths from (0, 0) to (n, n). Indeed, such parking functions, say $\{g^*(1),..., g^*(n)\}$, are the level code representations of such paths, specified by

$$g^*(1) = 1, \quad g^*(i-1) \leqslant g^*(i) \leqslant i, \quad i = 2(1)\,n,$$

which serves as a rapid way to write them down, and their permutations supply the remaining functions.

To see how (2.3) determines $g(1)$, given code $(c(1),..., c(n-1))$, consider first the function $\{h(1),..., h(n)\}$, with

$$\begin{aligned} h(1) &= n+1 \\ h(i+1) &= c(1) + \cdots + c(i), \bmod \overline{n+1}, \quad i = 1\,(1)\,\overline{n-1}. \end{aligned}$$

Take $r(h) = \{r_1,..., r_{n+1}\}$ as the specification of h (note that the range of h is $[n+1]$), in the sense given above, and let $R_j(h) = r_1 + \cdots + r_j - j$. Then there is a unique d such that $R_d(h) < R_j(h)$, $j = 1(1)\,\overline{d-1}$, and $R_d(h) \leqslant R_j(h)$, $j = \overline{d+1}(1)\,\overline{n+1}$ (d is the leftmost position of the minimums of $R_j(h)$). Note that $R_d(h) \leqslant -1$, since $r_1 + \cdots + r_{n+1} = n$, because h is a set with elements, and so $R_{n+1}(h) = -1$. This result and the preceding characterization of $R_d(h)$ entail $r_d = 0$; we do not take space for the proof.

Now suppose $g(1) = n+1-d$ and, as in (2.2),

$$g(i) = g(1) + c(1) + \cdots + c(i-1), \bmod \overline{n+1}, \quad i = 2\,(1)\,n$$

so that $g(i) = h(i) - d$, $i = 1(1)n$. The specification of $g = \{g(1),..., g(n)\}$ is clearly $(r_{d+1},..., r_{n+1}, r_1,..., r_d)$. Hence in the first place, $R_n(g) = r_1 + \cdots + r_{n+1} - r_d - n = 0$. Next

$$\begin{aligned} R_j(g) &= R_{d+j}(h) - R_d(h), & j &= 1\,(1)\,\overline{n+1-d} \\ &= R_{d+j-n-1}(h) - R_d(h) + R_{n+1}(h), & j &= \overline{n+2-d}\,(1)\,\overline{n+1}. \end{aligned}$$

These relations and the inequalities for $R_d(h)$ show that $R_j(g) \geqslant 0$, $j = 1(1)\,\overline{n-1}$. Hence g is a parking function when $g(1) = n+1-d$.

To complete the story we now give a brief description of the Prüfer code for trees (other codes appear in [2]) and its inverse. The code is obtained as follows: c_1 is the point adjacent to the endpoint of the tree with smallest label; c_2 is the point adjacent to the endpoint with smallest label of the tree truncated by removal of the first endpoint and its line to c_1; the process continues until a single line remains. For the inverse, following E. H. Neville [6], write down two sequences:

$$b_1, b_2, \ldots, b_{n-2}, b_{n-1}$$
$$c_1, c_2, \ldots, c_{n-2}, n.$$

The second of course is the code augmented by n. The first is the sequence of points adjacent to the code points in the process described above. Then b_1 is the smallest number not in $c_1, \ldots, c_{n-2}, n$; b_i, $i = 2(1)\overline{n-1}$, is the smallest number not in $b_1, \ldots, b_{i-1}$, $c_i, \ldots, c_{n-2}$, n. Since the pairs (b_i, c_i), $i = 1(1)\, n-2$, and (b_{n-1}, n) are $n-1$ lines of the tree, the tree is completely determined.

Figure 1 shows the mapping of all acyclic and parking functions for $n=3$, along with their codes and trees.

3. The Second Mapping

Let A_n and B_n denote respectively the sets of acyclic and parking functions on $[n]$. Another set C_n is introduced and will be put in one-to-one correspondence with both A_n and B_n. First we say that a sequence $r = (r_1, \ldots, r_n)$ of n non-negative integers is *balanced* if conditions (2.3) are fulfilled, i.e. if

$$r_1 + \cdots + r_j - j \geqslant 0, \quad j = 1(1)\overline{n-1}, \quad r_1 + \cdots + r_n - n = 0. \tag{3.1}$$

Throughout this section the sequence of the non-zero elements of r will be denoted by

$$(r_{i_1}, r_{i_2}, \ldots, r_{i_m}). \tag{3.2}$$

We have $1 \leqslant m \leqslant n$ and $1 = i_1 < i_2 < \cdots < i_m \leqslant n$. We also let

$$\mathbf{r}_{i_j} = r_{i_1} + \cdots + r_{i_j}, \quad j = 1(1)\, m.$$

The sequence r is then balanced if and only if

$$\mathbf{r}_{i_j} - (i_{j+1} - 1) \geqslant 0, \quad j = 1(1)\overline{m-1}, \quad \mathbf{r}_{i_m} - n = 0. \tag{3.3}$$

We also say that a permutation π of the set $[n]$ is *compatible* with the sequence r if the inverse π^{-1} of π is *increasing* on $\{1, \ldots, r_{i_1}\}$ and on each interval $\{\mathbf{r}_{i_j}+1, \ldots, \mathbf{r}_{i_{j+1}}\}$, $j = 1(1)\overline{m-1}$. Then C_n is defined as the set of all couples (r, π) with r a balanced sequence and π a permutation compatible with r.

Mapping B_n into C_n is easy. Let g be a parking function. The couple $(s(g), \tau_g)$ of C_n associated with g is defined as follows. First, $s(g)$ is simply the *specification* of g. From the previous section (relation (2.3)) we know that $s(g)$ is balanced. Secondly the permutation τ_g is defined by

$$\tau_g(x) = \text{Card}\{y \in [n] : g(y) < g(x), \quad \text{or} \quad g(y) = g(x) \quad \text{and} \quad y \leqslant x\}. \tag{3.4}$$

If $s(g) = r = (r_1, \ldots, r_n)$ and if we keep notations (3.2), then $\tau_g(x)$ may be described by: the numbers $1, 2, \ldots, r_{i_1}$ appear in the successive positions left to right where $g(x) = 1$; also, for any $j = 2(1)m$, the numbers $\mathbf{r}_{i_{j-1}} + 1, \ldots, \mathbf{r}_{i_j}$ appear in the similar positions where $g(x) = i_j$. This is again equivalent to saying that

$$\left.\begin{aligned} g(x) &= 1 \quad \text{if} \quad 1 \leqslant \tau_g(x) \leqslant r_{i_1} \\ &= i_j \quad \text{if} \quad \mathbf{r}_{i_{j-1}} + 1 \leqslant \tau_g(x) \leqslant \mathbf{r}_{i_j}, \quad j = 2(1)\, m \end{aligned}\right\} \tag{3.5}$$

and also that τ_g^{-1} is increasing on $\{1, \ldots, r_{i_1}\}$ and on $\{\mathbf{r}_{i_{j-1}} + 1, \ldots, \mathbf{r}_{i_j}\}$, $j = 2(1)m$. In particular τ_g is compatible with $s(g)$.

EXAMPLE. For $n = 3$, the functions g, specifications $s(g) = (r_1, r_2, r_3)$ and permutations τ_g are as follows

g	111	112	121	211	113	131	311	122	212	221
$s(g)$	300	210	210	210	201	201	201	120	120	120
τ_g	123	123	132	312	123	132	312	123	213	231

The remaining six g's are permutations, of specification (111), and $\tau_g = g$. Notice that, for each of the five ordered functions 111, 112, 113, 122, 123, $\tau_g = 123 = I$, the identity permutation.

Now let g and g' be two distinct parking functions. If $s(g) \neq s(g')$, then of course $(s(g), \tau_g) \neq (s(g'), \tau_{g'})$. But because of relations (3.5) we have the same conclusion when $s(g) = s(g')$. Hence the mapping $g \to (s(g), \tau_g)$ is *injective*.

To prove that the mapping $g \to (s(g), \tau_g)$ is *surjective* and at the same time define its reverse, we let (r, π) be an element of C_n. Taking again notations (3.2) for r we define the function g as follows:

$$\left.\begin{aligned} g(x) &= 1 \quad \text{if} \quad 1 \leqslant \pi(x) \leqslant r_{i_1} \\ &= i_j \quad \text{if} \quad \mathbf{r}_{i_{j-1}} + 1 \leqslant \pi(x) \leqslant \mathbf{r}_{i_j}, \quad j = 2(1)\, m\,. \end{aligned}\right\} \tag{3.6}$$

Clearly r is the specification of g and as r is balanced, if follows from (2.3) that g is a parking function. Furthermore, as π and τ_g are both compatible with r, relations (3.5) and (3.6) imply that they are equal. Thus $g \to (s(g), \tau_g)$ is a *bijection* of B_n onto C_n.

We now map A_n into C_n. The couple of C_n associated with an acyclic function f is denoted by $(t(f), \sigma_f)$. The permutation σ_f seems most easily stated in terms of the forest for f. It is also convenient to use the labels $1, 2, \ldots, n$ as names of the vertices of this forest. If $f(x)=x'$ and $x \neq x'$, there is just one line in the direct path going from x to x'. We then say that x is *at height* 1 *from* x'. A vertex x belongs to a particular connected component of the forest, which is a tree of root say z. The number of lines in the direct path going from x to z is the *height* of x. The roots are of course of height zero. A *total order* denoted by $\leqslant_f$ is now defined on the set $[n]$ (of course $x <_f y$ means $x \leqslant_f y$ and $x \neq y$). First we let $x <_f y$ if the height of x is less than the height of y or if x and y are both roots and $x < y$. Then assume that the order $\leqslant_f$ has been defined on the set of all vertices of height $\leqslant k$ $(k \geqslant 0)$. If x and y are two vertices of height $k+1$, the points $f(x)$ and $f(y)$ are both of height k. By induction we let

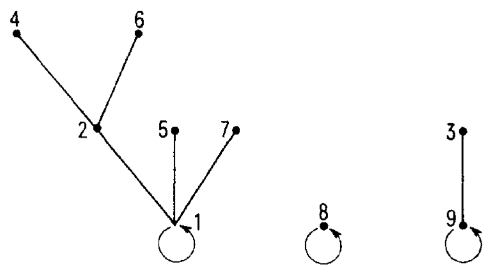
$$x <_f y \quad \text{if} \quad \left.\begin{array}{ll} \text{either} & f(x) <_f f(y) \\ \text{or} & f(x) = f(y) \quad \text{and} \quad x < y . \end{array}\right\} \tag{3.7}$$

The sequence formed by writing the n elements of $[n]$ in *increasing* order with respect to $\leqslant_f$ is denoted by $(\sigma_f^{-1}(1), \ldots, \sigma_f^{-1}(n))$. Of course σ_f is the inverse permutation of σ_f^{-1}.

EXAMPLE. Take $n=9$ and

$$\begin{array}{rccccccccc} x= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ f(x)= & 1 & 1 & 9 & 2 & 1 & 2 & 1 & 8 & 9. \end{array}$$

With the ordering described above, the associated forest is



and

$$\begin{array}{rccccccccc} \sigma_f^{-1}(x) = & 1 & 8 & 9 & 2 & 5 & 7 & 3 & 4 & 6 \\ \sigma_f(x) = & 1 & 4 & 7 & 8 & 5 & 9 & 6 & 2 & 3. \end{array}$$

Now define the *forest-specification* of f as the sequence $t(f)=(r_1, \ldots, r_n)$ with r_1 the number of roots and r_i the number of vertices at height 1 from $\sigma_f^{-1}(i-1)$ for $i=2(1)n$. For instance the forest-specification in the above example is $t(f)$

$=(3, 3, 0, 1, 2, 0, 0, 0, 0)$. Let $(r_{i_1}, r_{i_2}, \ldots, r_{i_m})$ be the sequence of the non zero elements of $t(f)$. We have

$$\mathbf{r}_{i_m} = r_{i_1} + \cdots + r_{i_m} = n \tag{3.8}$$

since there are n lines or slings in a forest with n labeled vertices. On the other hand the first $r_{i_1}(=r_1)$ elements, $\sigma_f^{-1}(1), \ldots, \sigma_f^{-1}(r_{i_1})$, are the r_{i_1} roots in rising order. Thus

$$f(x) = x \quad \text{if} \quad 1 \leqslant \sigma_f(x) \leqslant r_{i_1}. \tag{3.9}$$

Moreover, for $j=2(1)m$ the elements $\sigma_f^{-1}(\mathbf{r}_{i_{j-1}}+1), \ldots, \sigma_f^{-1}(\mathbf{r}_{i_j})$ are the vertices at height 1 from $\sigma_f^{-1}(i_j-1)$ also written in rising order because of (3.7). Hence

$$f(x) = \sigma_f^{-1}(i_j - 1) \quad \text{if} \quad \mathbf{r}_{i_{j-1}} + 1 \leqslant \sigma_f(x) \leqslant \mathbf{r}_{i_j}, \quad j = 2(1)\,m. \tag{3.10}$$

Also

$$i_j - 1 < \mathbf{r}_{i_{j-1}} + 1, \tag{3.11}$$

i.e.

$$\mathbf{r}_{i_{j-1}} + 1 - (i_j - 1) > 0, \quad j = 2(1)\,m. \tag{3.12}$$

since a vertex of smaller height precedes a vertex of greater height in the sequence $(\sigma_f^{-1}(1), \ldots, \sigma_f^{-1}(n))$. Relations (3.8) and (3.12) imply that the sequence $t(f)=(r_1, \ldots, r_n)$ is balanced. As we have also noted that σ_f^{-1} is increasing on $\{1, \ldots, r_{i_1}\}$ and on each interval $\{\mathbf{r}_{i_{j-1}}+1, \ldots, \mathbf{r}_{i_j}\}, j=2(1)m$, the permutation σ_f is compatible with $t(f)$. Thus $f \to (t(f), \sigma_f)$ *maps* A_n *into* C_n.

Furthermore let f and f' be two distinct acyclic functions. If $\sigma_f = \sigma_{f'}$, then relations (3.9) and (3.10) immediately show that $(t(f), \sigma_f) \neq (t(f'), \sigma_{f'})$. Trivially the same conclusion also holds when $\sigma_f \neq \sigma_{f'}$. Hence $f \to (t(f), \sigma_f)$ is *injective*.

Finally let us prove that $f \to (t(f), \sigma_f)$ is *surjective* and define its inverse mapping from C_n onto A_n. Let (r, π) be an element of C_n and let again $(r_{i_1}, r_{i_2}, \ldots, r_{i_m})$ be the sequence of the non zero elements of r. As all the r_{i_j}'s are positive and since $\mathbf{r}_{i_m} = r_{i_1} + \ldots + r_{i_m} = n$, we can define a function of $[n]$ into itself by

$$\left.\begin{aligned} f(x) &= x \quad \text{if} \quad 1 \leqslant \pi(x) \leqslant r_{i_1} \\ &= \pi^{-1}(i_j - 1) \quad \text{if} \quad \mathbf{r}_{i_{j-1}} + 1 \leqslant \pi(x) \leqslant \mathbf{r}_{i_j}, \quad j = 2(1)\,m. \end{aligned}\right\} \tag{3.13}$$

As r is balanced, relations (3.11) hold for $j=2(1)m$. Hence the function f defined by (3.13) is necessarily an acyclic function of roots $\pi^{-1}(1), \ldots, \pi^{-1}(r_{i_1})$. It remains to show that $(t(f), \sigma_f) = (r, \pi)$. As π is compatible with r, the definition of the total order $\leqslant_f$ together with (3.13) implies that $\pi^{-1}(1) <_f \cdots <_f \pi^{-1}(r_{i_1})$ and $\pi^{-1}(\mathbf{r}_{i_{j-1}}+1) <_f \cdots <_f \pi^{-1}(\mathbf{r}_{i_j}), j=2(1)m$. On the other hand $\pi^{-1}(r_{i_1})$ is of height zero and $\pi^{-1}(r_{i_1}+1)$ is of height 1. This gives the inequality $\pi^{-1}(r_{i_1}) <_f \pi^{-1}(r_{i_1}+1)$. Assume that $\pi^{-1}(\mathbf{r}_{i_{k-1}}) <_f \pi^{-1}(\mathbf{r}_{i_{k-1}}+1)$ holds for all k such that $2 \leqslant k \leqslant j < m$. This is equivalent to assuming that $\pi^{-1}(1) <_f \pi^{-1}(2) <_f \ldots <_f \pi^{-1}(\mathbf{r}_{i_j})$. The two elements $\pi^{-1}(\mathbf{r}_{i_j})$ and $\pi^{-1}(\mathbf{r}_{i_j}+1)$ are at height 1 from $\pi^{-1}(i_j-1)$ and $\pi^{-1}(i_{j+1}-1)$ respectively. From (3.11) we have

$$i_j - 1 < i_{j+1} - 1 \leqslant \mathbf{r}_{i_j}.$$

By induction on j, this implies

$$\pi^{-1}(i_j - 1) <_f \pi^{-1}(i_{j+1} - 1)$$

and also

$$\pi^{-1}(\mathbf{r}_{i_j}) = f\pi^{-1}(i_j - 1) <_f f\pi^{-1}(i_{j+1} - 1) = \pi^{-1}(\mathbf{r}_{i_j} + 1)$$

by definition of the total order $\leqslant_f$. Thus $\sigma_f = \pi$. This in its turn, together with the definition of f in (3.13), implies that $t(f) = r$. Thus $f \to (t(f), \sigma_f)$ is a bijection of A_n onto C_n.

Let f be an acyclic function. Put $(r, \pi) = (t(f), \sigma_f)$ and let g be defined by (3.6). Then $f \to g$ is a bijection of A_n onto B_n and is our second mapping. Furthermore it follows from formulas (3.9), (3.10) and (3.6) that g can be expressed by

$$\left.\begin{aligned} g(x) &= 1 \quad \text{if} \quad f(x) = x \\ &= 1 + \sigma_f(f(x)) \quad \text{if} \quad f(x) \neq x. \end{aligned}\right\} \tag{3.14}$$

In the same way we have an explicit formula for the reverse mapping $g \to f$ from B_n onto A_n. Let $(r, \pi) = (s(g), \tau_g)$. Then formulas (3.5) and (3.13) imply that

$$\left.\begin{aligned} f(x) &= x \quad \text{if} \quad g(x) = 1 \\ &= \tau_g^{-1}(g(x) - 1) \quad \text{if} \quad g(x) > 1. \end{aligned}\right\} \tag{3.15}$$

EXAMPLE. The mappings for $n=3$ are as follows

f	111	222	333	112	131	221	313	322	233
$t(f) = s(g)$	120	120	120	111	111	111	111	111	111
$\sigma_f = \tau_g$	123	213	231	123	132	213	231	312	321
g	122	212	221	123	132	213	231	312	321

f	122	133	323	121	113	223	123
$t(f) = s(g)$	201	201	201	210	210	210	300
$\sigma_f = \tau_g$	123	213	231	123	132	312	123
g	113	131	311	112	121	211	111

It is clear from this that the ordered parking function with specification

$r=(r_{i_1}, r_{i_2}, \ldots, r_{i_m})$ is mapped with the acyclic function whose forest-specification is equal to r and points labeled in increasing sequence: roots labeled $1, \ldots, r_{i_1}$, points $\mathbf{r}_{i_{j-1}}+1, \ldots, \mathbf{r}_{i_j}$ at height 1 from i_j-1, $j=2(1)m$.

4. Some Uses of The Mappings

To show that our mappings have point, we give here a number of examples of their implications.

In the first mapping, a zero ($\equiv n+1 \bmod \overline{n+1}$) in the code indicates a pair of consecutive like numbers in the parking function and also the codes with k zeroes correspond to acyclic functions with $k+1$ fixed points. Hence, using [8],

PROPOSITION 1. *The enumerator of parking functions by number of pairs of like consecutive numbers is* $x^{-1}A_n(x)=(x+n)^{n-1}$.

The second mapping shows that acyclic functions with k fixed points are equinumerous with parking functions with k elements equal to 1 (that is $r_1=k$ in the specification $(r_1, \ldots, r_n)$). Hence, again using [8],

PROPOSITION 2. *The enumerator of parking functions by number of elements equal to* 1 *is* $A_n(x)=x(x+n)^{n-1}$.

The second mapping also shows that ordered parking functions with n elements are equinumerous with plane rooted trees with $n+1$ unlabeled points, a result familiar from Harris [4]. The number of both is a_{nn}. Refining the equivalence leads to

PROPOSITION 3. *The enumerator of rooted plane trees with* $n+1$ *like points by number of branches at the root is*

$$\sum_{1}^{n} a_{n-1,n-j}x^j,$$

with a_{nk} *the ballot number as already defined.*

Proposition 2 may be used for a further result as follows. Write

$$P_n(x, y) = \sum_{i,j=1}^{n} P_{ij}(n)\, x^i y^j = \sum_{i=1}^{n} x^i p_{ni}(y)$$

for the enumeraotr of parking functions on $[n]$ by leading element and by 'number of ones'; that is, $P_{ij}(n)$ is the number of functions with $g(1)=i$ and card $\{x\colon g(x)=1\}=j$. The second expression is of course a definition of $p_{ni}(y)$, the enumerator of functions with leading element i by number of ones.

Then, first $p_{nn}(y)=A_{n-1}(y)$, since with $g(1)=n$, the remaining g's are restricted to set $[n-1]$ and by proposition 2 are enumerated by $A_{n-1}(y)$. Next

$$p_{n,n-1}(y) = p_{nn}(y) + (n-1) A_1(1) A_{n-2}(y).$$

In the first term, the choices for $\{g(2),\dots,g(n)\}$ are in $[n-1]$; in the second they are in the disjoint sets $[n-2]$ and $\{n\}$.

A similar argument shows that

$$p_{n,n-k}(y) = p_{n,n-k+1}(y) + \binom{n-1}{k} A_k(1) A_{n-k-1}(y), \quad k = 1(1)\, n-2. \tag{4.1}$$

Indeed, the choices are divided into those from the disjoint sets $[n-k-1]$ and $\{n, n-1,\dots, n-k+1\}$, and those which are not.

Finally

$$p_{n1}(y) = y(A_{n-1}(1) + p_{n2}(y)). \tag{4.2}$$

Iteration of (4.1) leads to

$$p_{nk}(y) = \sum_{j=0}^{n-k} \binom{n-1}{j} A_j(1) A_{n-1-j}(y), \quad k = 2(1)\, n. \tag{4.3}$$

Hence, by (4.2),

$$p_{n1}(y) = y \sum_{j=0}^{n-1} \binom{n-1}{j} A_j(1) A_{n-1-j}(y) = yA_{n-1}(1+y). \tag{4.4}$$

The last relation follows from the binomial relation

$$A_n(x+y) = \sum_0^n \binom{n}{k} A_k(x) A_{n-k}(y)$$

which is implicit in [8].

From (4.3) and (4.4), $P_n(x, y)$ is found to be

$$P_n(x, y) = \sum_0^{n-1} \binom{n-1}{j} a_j(x, y) A_{n-1-j}(1) A_j(y) \tag{4.5}$$

with

$$a_0(x, y) = xy, \quad a_j(x, y) = x(y + x + \cdots + x^j), \quad j = 1, 2, \dots.$$

Hence,

$$\left.\begin{aligned}(1-x)x^{-1}P(x, y) &= \sum_0^{n-1} \binom{n-1}{j} (x + y - xy - x^{j+1}) \\ &\quad \times A_{n-1-j}(1) A_j(y) \\ &= (x + y - xy) A_{n-1}(1+y) \\ &\quad - \sum_0^{n-1} \binom{n-1}{j} x^{j+1} A_{n-1-j}(1) A_j(y)\end{aligned}\right\} \tag{4.6}$$

and the enumerator of parking functions by lead numbers is

$$(1-x)\,x^{-1}P_n(x,1) = 2(n+1)^{n-2} - \sum_{0}^{n-1} \binom{n-1}{j} x^j T_{n-j} T_{j+1}. \tag{4.7}$$

The first few values are $P_2(x,1)=2x+x^2$, $P_3(x,1)=8x+5x^2+3x^3$, $P_4(x,1)=50x+34x^2+25x^3+16x^4$.

REFERENCES

[1] Comtet, L., *Analyse Combinatoire, Vol. 1* (Presses Universitaires de France, Paris 1970).
[2] Foata, D. et Fuchs, A., *Réarrangements de fonctions et dénombrement*, J. Combinatorial Theory *8*, 361–375 (1970).
[3] Françon, J., *Scatter Storage and Trees* (submitted for publication in Comm. A.C.M.).
[4] Harris, T. E., *First Passage and Recurrence Distributions*, Trans. Amer. Math. Soc. *73*, 471–486 (1952).
[5] Konheim, A. G. and Weiss, B., *An Occupancy Discipline and Applications*, SIAM J. Appl. Math. *14*, 1266–1274 (1966).
[6] Neville, E. H., *The Codifying of Tree Structure*, Proc. Cambridge Philos. Soc. *49*, 381–385 (1953).
[7] Riordan, J., *Enumeration of Linear Graphs for Mappings of Finite Sets*, Ann. Math. Statist. *33*, 178–185 (1962).
[8] Riordan, J., *Forests of Labeled Trees*, J. Combinatorial Theory *5*, 90–103 (1968).
[9] Riordan, J., *Ballots and Trees*, J. Combinatorial Theory *6*, 408–411 (1969).
[10] Schützenberger, M. P., *On an Enumeration Problem*, J. Combinatorial Theory *4*, 219–221 (1968).

University of Florida[2])
Rockefeller University

[2]) On leave from the Université de Strasbourg, 1971–1972.