A Determinant of Stirling Cycle Numbers Counts Unlabeled Acyclic Single-Source Automata

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [arXiv:0704.0004v1 [math.CO] 31 Mar 2007](http://arxiv.org/abs/0704.0004v1) | DAVID CALLAN  Department of Statistics  University of Wisconsin-Madison  1300 University Ave  Madison, WI 53706-1532  callan@stat.wisc.edu  March 30, 2007  Abstract  We show that a determinant of Stirling cycle numbers counts unlabeled acyclic | | | | | | | | | | | | | | | | |
| single-source automata. | | | The proof involves a bijection from these automata to | | | | | | | | | | | | | |
| certain marked lattice paths and a sign-reversing involution to evaluate the deter-minant. | | | | | | | | | | | | | | | | |
| 1 | | Introduction The chief purpose of this paper is to show bijectively that | | | | | | | | | | | | | | |
| a determinant of Stirling cycle numbers counts unlabeled acyclic single-source automata. | | | | | | | | | | | | | | | | |
| Specifically, let Ak(n) denote the kn × kn matrix with (i, j) entry | | | | | | | | | | | | | | | ⌊i−1 k⌋+2 | , where |
| ⌊i−1 k⌋+1+i−j |
| i  j | is the Stirling cycle number, the number of permutations on [i] with j cycles. For | | | | | | | | | | | | | | | |
| example, | | | | | | | | | | | | | | | | |
| A2(5) = | | |    | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |    | . | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 6 | 11 | 6 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 6 | 11 | 6 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 10 | 35 | 50 | 24 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 | 35 | 50 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 15 | 85 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 15 |

1

As evident in the example, Ak(n) is formed from k copies of each of rows 2 through n + 1 of the Stirling cycle triangle, arranged so that the first nonzero entry in each row is a 1 and, after the first row, this 1 occurs just before the main diagonal; in other words, Ak(n) is a Hessenberg matrix with 1s on the infra-diagonal. We will show

Main Theorem. The determinant of Ak(n) is the number of unlabeled acyclic single-source automata with n transient states on a (k + 1)-letter input alphabet.

Section 2 reviews basic terminology for automata and recurrence relations to count finite acyclic automata. Section 3 introduces column-marked subdiagonal paths, which play an intermediate role, and a way to code them. Section 4 presents a bijection from these column-marked subdiagonal paths to unlabeled acyclic single-source automata. Fi-nally, Section 5 evaluates det Ak(n) using a sign-reversing involution and shows that the determinant counts the codes for column-marked subdiagonal paths.

2 Automata

A (complete, deterministic) automaton consists of a set of states and an input alphabet whose letters transform the states among themselves: a letter and a state produce another state (possibly the same one). A finite automaton (finite set of states, finite input alphabet of, say, k letters) can be represented as a k-regular directed multigraph with ordered edges: the vertices represent the states and the first, second, . . . edge from a vertex give the effect of the first, second, . . . alphabet letter on that state. A finite automaton cannot be acyclic in the usual sense of no cycles: pick a vertex and follow any path from it. This path must ultimately hit a previously encountered vertex, thereby creating a cycle. So the term acyclic is used in the looser sense that only one vertex, called the sink, is involved in cycles. This means that all edges from the sink loop back to itself (and may safely be omitted) and all other paths feed into the sink.

A non-sink state is called transient. The size of an acyclic automaton is the number of transient states. An acyclic automaton of size n thus has transient states which we label 1, 2, . . . , n and a sink, labeled n + 1. Liskovets [1] uses the inclusion-exclusion principle (more about this below) to obtain the following recurrence relation for the number ak(n)

2

of acyclic automata of size n on a k-letter input alphabet (k ≥1):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ak(0) = 1; | ak(n) = | n−1 | (−1)n−j−1 | n | (j + 1)k(n−j)ak(j), | n ≥1. |
| j=0 | j |

A source is a vertex with no incoming edges. A finite acyclic automaton has at least

one source because a path traversed backward v1 ←v2 ←v3 ←. . . must have distinct

vertices and so cannot continue indefinitely. An automaton is single-source (or initially

connected) if it has only one source. Let Bk(n) denote the set of single-source acyclic

finite (SAF) automata on a k-letter input alphabet with vertices 1, 2, . . . , n + 1 where 1

is the source and n + 1 is the sink, and set bk(n) = | Bk(n) |. The two-line representation

of an automaton in Bk(n) is the 2 × kn matrix whose columns list the edges in order. For

example,

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B = | | | | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | c →3 | a →6, 1 | b → |
| 2 | 4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 3 | 5 | 3 | 2 | 2 | 6 |
| is in B3(5) and the source-to-sink paths in B include 1 | | | | | | | | | | | | | | a →2 | a →6, 1 | | | b →4 |
| 4 | b →5 | b →2 | b →6, where the alphabet is {a, b, c}. | | | | | | | | | | |

Proposition 1. The number bk(n) of SAF automata of size n on a k-letter input alphabet

(n, k ≥1) is given by

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Remark | bk(n) = | n | (−1)n−i | n −1 | (i + 1)k(n−i)ak(i) |
| i=1 | i −1 |
| This formula is a bit more succinct than the the recurrence in [1, Theorem | | | | |

3.2].

Proof Consider the set A of acyclic automata with transient vertices [n] = {1, 2, . . . , n}

in which 1 is a source. Call 2, 3, . . . , n the interior vertices. For X ⊆[2, n], let

f(X) = # automata in A whose set of interior vertices includes X,

g(X) = # automata in A whose set of interior vertices is precisely X.

Then f(X) = Y : X⊆Y ⊆[2,n]g(Y ) and by M¨obius inversion [2] on the lattice of subsets of

[2, n], g(X) = Y : X⊆Y ⊆[2,n]µ(X, Y )f(Y ) where µ(X, Y ) is the M¨obius function for this

3

lattice. Since µ(X, Y ) = (−1)|Y |−|X|if X ⊆Y , we have in particular that

|  |  |  |
| --- | --- | --- |
| g(∅) = | (−1)| Y |f(Y ). | (1) |

Y ⊆[2,n]

Let | Y | = n −i so that 1 ≤i ≤n. When Y consists entirely of sources, the vertices in [n + 1]\Y and their incident edges form a subautomaton with i transient states; there are ak(i) such. Also, all edges from the n −i vertices comprising Y go directly into [n + 1]\Y : (i + 1)k(n−i)choices. Thus f(Y ) = (i + 1)k(n−i)ak(i). By definition, g(∅) is the number of automata in A for which 1 is the only source, that is, g(∅) = bk(n) and the Proposition now follows from (1).

An unlabeled SAF automaton is an equivalence class of SAF automata under relabeling of the interior vertices. Liskovets notes [1] (and we prove below) that Bk(n) has no nontrivial automorphisms, that is, each of the (n −1)! relabelings of the interior vertices of B ∈Bk(n) produces a different automaton. So unlabeled SAF automata of size n on a k-letter alphabet are counted by (n−1)!bk(n). The next result establishes a canonical 1

representative in each relabeling class.

Proposition 2. Each equivalence class in Bk(n) under relabeling of interior vertices has size (n −1)! and contains exactly one SAF automaton with the “last occurrences increas-ing” property: the last occurrences of the interior vertices—2, 3, . . ., n—in the bottom row of its two-line representation occur in that order.

Proof The first assertion follows from the fact that the interior vertices of an au-tomaton B ∈bk(n) can be distinguished intrinsically, that is, independent of their labeling. To see this, first mark the source, namely 1, with a mark (new label) v1 and observe that there exists at least one interior vertex whose only incoming edge(s) are from the source (the only currently marked vertex) for otherwise a cycle would be present. For each such interior vertex v, choose the last edge from the marked vertex to v using the built-in ordering of these edges. This determines an order on these vertices; mark them in order v2, v3, . . . , vj (j ≥2). If there still remain unmarked interior vertices, at least one of them has incoming edges only from a marked vertex or again a cycle would be present. For each such vertex, use the last incoming edge from a marked vertex, where now edges are arranged in order of initial vertex vi with the built-in order breaking ties, to order and

4

mark these vertices vj+1, vj+2, . . .. Proceed similarly until all interior vertices are marked. For example, for

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B = | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | , |
| 2 | 4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 3 | 5 | 3 | 2 | 2 | 6 |

v1 = 1 and there is just one interior vertex, namely 4, whose only incoming edge is from the source, and so v2 = 4 and 4 becomes a marked vertex. Now all incoming edges to both 3 and 5 are from marked vertices and the last such edges (built-in order comes into

|  |  |  |
| --- | --- | --- |
| play) are 4 | b →5 and 4 | c →3 putting vertices 3, 5 in the order 5, 3. So v3 = 5 and v4 = 3. |

Finally, v5 = 2. This proves the first assertion. By construction of the vs, relabeling each interior vertex i with the subscript of its corresponding v produces an automaton in Bk(n) with the “last occurrences increasing” property and is the only relabeling that does so.

The example yields

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | . |
| 5 | 2 | 6 | 4 | 3 | 4 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

Now let Ck(n) denote the set of canonical SAF automata in Bk(n) representing un-

|  |  |
| --- | --- |
| labeled automata; thus | Ck(n) | = | 1  (n−1)!bk(n). Henceforth, we identify an unlabeled au- |

tomaton with its canonical representative.

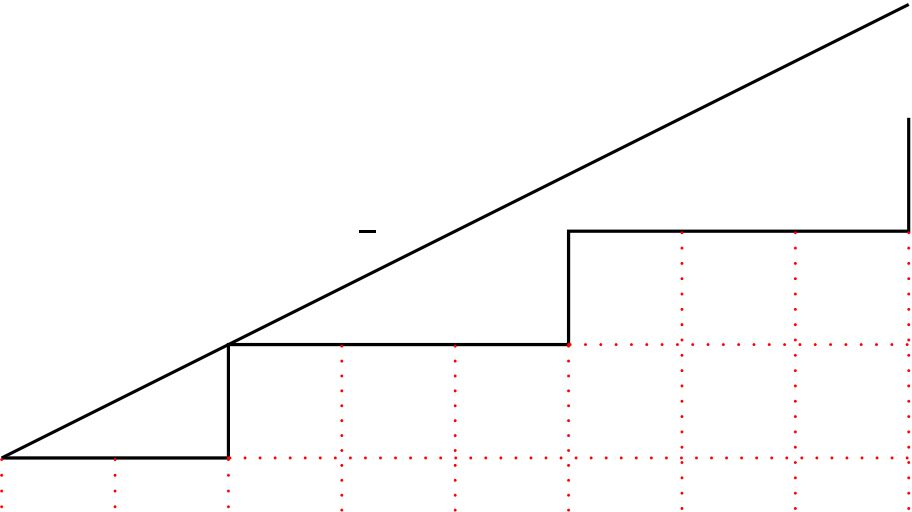
3 Column-Marked Subdiagonal Paths

A subdiagonal (k, n, p)-path is a lattice path of steps E = (1, 0) and N = (0, 1), E for east and N for north, from (0, 0) to (kn, p) that never rise above the line y =1 kx. Let Ck(n, p) denote the set of such paths.For k ≥1, it is clear that Ck(n, p) is nonempty only for 0 ≤p ≤n and it is known (generalized ballot theorem) that

|  |  |  |
| --- | --- | --- |
| | Ck(n, p) | =kn −kp + 1 kn + p + 1 | kn + p + 1 | . |
| p |

A path P in Ck(n, n) can be coded by the heights of its E steps above the line y = −1; this gives a a sequence (bi)kn i=1subject to the restrictions 1 ≤b1≤b2≤. . . ≤bknand bi ≤⌈i/k⌉for all i.

5



A column-marked subdiagonal (k, n, p)-path is one in which, for each i ∈[1, kn], one of

the lattice squares below the ith E step and above the horizontal line y = −1 is marked,

∗  
say with a ‘ ∗’. Let C k(n, p) denote the set of such marked paths.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| y =1 2x | b | b | b | b | b | (8,4) |
| b |
| b | b |

∗

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (0,0) | b | ∗ | b | ∗ | b | ∗ | b | ∗ | b | ∗ | b | ∗ | ∗ |
| b |

y = −1

∗  
A path in C 2(4, 3)

∗ kn A marked path P∗in C k(n, n) can be coded by a sequence of pairs (ai, bi) i=1where

(bi)kn i=1is the code for the underlying path P and ai ∈[1, bi] gives the position of the ∗in the

ith column. The example is coded by (1, 1), (1, 1), (1, 2), (2, 2), (1, 2), (3, 3), (1, 3), (2, 3).

∗  
An explicit sum for | C k(n, n) | is

|  |  |
| --- | --- |
| ∗ | C k(n, n) | = | b1b2 . . . bkn, |

1≤b1≤b2≤...≤bkn,

bi ≤⌈i/k⌉for all i

because the summand b1b2 . . . bkn is the number of ways to insert the ‘ ∗’s in the underlying

path coded by (bi)kn i=1.

∗  
It is also possible to obtain a recurrence for | C k(n, p) |, and then, using Prop. 1, to

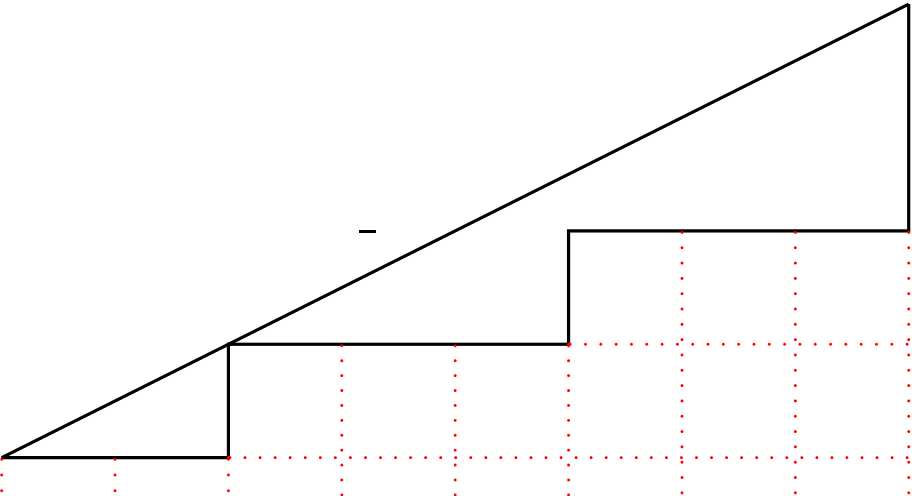
∗  
show analytically that | C k(n, n) | = | Ck+1(n) |. However, it is much more pleasant to

give a bijection and in the next section we will do so. In particular, the number of SAF

automata on a 2-letter alphabet is

∗  
| C2(n) | = | C 1(n, n) | =

1≤b1≤b2≤...≤bn   
bi ≤i for all i



sequence [A082161](http://www.research.att.com:80/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A082161) in [3].

4 Bijection from Paths to Automata

∗

In this section we exhibit a bijection from C k(n, n) to Ck+1(n). Using the illustrated

path as a working example with k = 2 and n = 4,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| y =1 2x | b | b | b | b | b | (8,4) |
| b |
| b | b |

∗

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (0,0) | b | ∗ | b | ∗ | b | ∗ | b | ∗ | b | ∗ | b | ∗ | ∗ |
| b |

y = −1

first construct the top row of a two-line representation consisting of k + 1 each 1s, 2s,

. . . , n s and number them left to right:

1 1 1 2 2 2 3 3 3 4 4 4   
 1 2 3 4 5 6 7 8 9 10 11 12

 .

The last step in the path is necessarily an N step. For the second last, third last,. . . N steps

in the path, count the number of steps following it. This gives a sequence i1, i2, . . . , in−1

satisfying 1 ≤i1 < i2 < . . . < in−1 and ij ≤(k + 1)j for all j. Circle the positions

i1, i2, . . . , in−1 in the two-line representation and then insert (in boldface) 2, 3, . . . , n in

the second row in the circled positions:

⃝ 2 3 4⃝ 6 7 8 ⃝ 10 11 12  
2 1 1 1 2

3   
2 2 3 3

4 3 4 4 4 .

These will be the last occurrences of 2, 3, . . . , n in the second row. Working from the last

column in the path back to the first, fill in the blanks in the second row left to right as

follows. Count the number of squares from the ∗up to the path (including the ∗square)

7

and add this number to the nearest boldface number to the left of the current blank entry (if there are no boldface numbers to the left, add this number to 1) and insert the result in the current blank square. In the example the numbers of squares are 2,3,1,2,1,2,1,1 yielding  
⃝ 2 3 4⃝ 6 7 8 ⃝ 10 11 12

This will fill all blank entries except the last. Note that ∗s in the bottom row correspond  2 1 1 4 1 5 2 3 3 2 2 5 3 4 3 5 4 3 4 5 4 5 4  .

to sink (that is, n+1) labels in the second row. Finally, insert n+1 into the last remaining blank space to give the image automaton:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | . |
| 2 | 4 | 5 | 3 | 3 | 5 | 4 | 5 | 4 | 5 | 5 | 5 |

This process is fully reversible and the map is a bijection.

5 Evaluation of det Ak(n)

For simplicity, we treat the case k = 1, leaving the generalization to arbitrary k as a not-too-difficult exercise for the interested reader. Write A(n) for A1(n). Thus i+1   
A(n) = 2i−j 1≤i,j≤n. From the definition of det A(n) as a sum of signed products, we show that det A(n) is the total weight of certain lists of permutations, each list carrying weight ±1. Then a weight-reversing involution cancels all −1 weights and reduces the problem to counting the surviving lists. These surviving lists are essentially the codes for ∗  
paths in C 1(n, p), and the Main Theorem follows from §4.

To describe the permutations giving a nonzero contribution to det A(n) = σsgn σ ×n   
i=1ai,σ(i), define the code of a permutation σ on [n] to be the list c = (ci)n i=1with i+1   
ci = σ(i)−(i−1). Since the (i, j) entry of A(n), 2i−j   
 , is 0 unless j ≥i−1, we must haveσ(i) ≥i−1 for all i. It is well known that there are 2n−1such permutations, corresponding to compositions of n, with codes characterized by the following four conditions: (i) ci ≥0 for all i, (ii) c1 ≥1, (iii) each ci ≥1 is immediately followed by ci −1 zeros in the list, (iv)n i=1ci = n. Let us call such a list a padded composition of n: deleting the zeros is a bijection to ordinary compositions of n. For example, (3, 0, 0, 1, 2, 0) is a padded

8

composition of 6. For a permutation σ with padded composition code c, the nonzero

entries in c give the cycle lengths of σ. Hence sgn σ, which is the parity of “n −# cycles in σ”, is given by (−1)# 0s in c.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| We have det A(n) = | σsgn σn i=1ai,σ(i) = | | σsgn σn i=1 | | i+1  2i−σ(i) | , and so | (2) |
| det A(n) = | (−1)# 0s in c | n | i + 1 | |
| i=1 | i + 1 −ci | |
| c | |

where the sum is restricted to padded compositions c of n with ci ≤i for all i ([A002083](http://www.research.att.com:80/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A002083))

|  |  |  |
| --- | --- | --- |
| because | i+1 | = 0 unless ci ≤i. |
| i+1−ci |

Henceforth, let us write all permutations in standard cycle form whereby the smallest

entry occurs first in each cycle and these smallest entries increase left to right. Thus,

with dashes separating cycles, 154-2-36 is the standard cycle form of the permutation

(1 2 3 4 5 6 5 2 6 1 4 3). We define a nonfirst entry to be one that does not start a cycle. Thus the

preceding permutation has 3 nonfirst entries: 5,4,6. Note that the number of nonfirst

entries is 0 only for the identity permutation. We denote an identity permutation (of any

size) by ǫ.

By definition of Stirling cycle number, the product in (2) counts lists (πi)n i=1of permu-

tations where πi is a permutation on [i + 1] with i + 1 −ci cycles, equivalently, with ci ≤i nonfirst entries. So define Ln to be the set all lists of permutations π = (πi)n i=1where πi is a permutation on [i + 1], # nonfirst entries in πi is ≤i, π1 is the transposition (1,2),

each nonidentity permutation πi is immediately followed by ci −1 ǫ’s where ci ≥1 is the

number of nonfirst entries in πi (so the total number of nonfirst entries is n). Assign a weight to π ∈Ln by wt(π) = (−1)# ǫ’s in π. Then

det A(n) = wt(π).

π∈Ln

We now define a weight-reversing involution on (most of) Ln. Given π ∈Ln, scan the

list of its component permutations π1 = (1, 2), π2, π3, . . . left to right. Stop at the first

one that either (i) has more than one nonfirst entry, or (ii) has only one nonfirst entry, b

say, and b > maximum nonfirst entry m of the next permutation in the list. Say πk is the

permutation where we stop.

9

In case (i) decrement (i.e. decrease by 1) the number of ǫ’s in the list by splitting πk into two nonidentity permutations as follows. Let m be the largest nonfirst entry of πk and let ℓbe its predecessor. Replace πk and its successor in the list (necessarily an ǫ) by the following two permutations: first the transposition (ℓ, m) and second the permutation obtained from πk by erasing m from its cycle and turning it into a singleton. Here are two examples of this case (recall permutations are in standard cycle form and, for clarity, singleton cycles are not shown).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | 6 | → | i | 1 | 2 | 3 | 4 | 5 | 6 |
| πi | 12 | 13 | 23 | 14-253 | ǫ | ǫ | πi | 12 | 13 | 23 | 25 | 14-23 | ǫ |

and

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | 6 | → | i | 1 | 2 | 3 | 4 | 5 | 6 |
| πi | 12 | 23 | 14 | 13-24 | ǫ | 23 | πi | 12 | 23 | 14 | 24 | 13 | 23 |

The reader may readily check that this sends case (i) to case (ii).

In case (ii), πk is a transposition (a, b) with b > maximum nonfirst entry m of πk+1. In this case, increment the number of ǫ’s in the list by combining πk and πk+1 into a single permutation followed by an ǫ: in πk+1, b is a singleton; delete this singleton and insert b immediately after a in πk+1 (in the same cycle). The reader may check that this reverses the result in the two examples above and, in general, sends case (ii) to case (i). Since the map alters the number of ǫ’s in the list by 1, it is clearly weight-reversing. The map fails only for lists that both consist entirely of transpositions and have the form

|  |  |  |  |
| --- | --- | --- | --- |
| (a1, b1), (a2, b2), . . . , (an, bn) | with b1 ≤b2 ≤. . . ≤bn. | | n  i=1satisfying |
| Such lists have weight 1. Hence det A(n) is the number of lists | | (ai, bi) |

1 ≤ai < bi ≤i + 1 for 1 ≤i ≤n, and b1 ≤b2 ≤. . . ≤bn. After subtracting 1 from each ∗ ∗  
bi, these lists code the paths in C 1(n, n) and, using §4, det A(n) = | C 1(n, n) | = | C2(n) |.

References

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [1] Valery | A. | Liskovets, | | Exact | in | enumeration | | of | acyclic | deterministic | | au- |
| tomata, | Disc. | | Appl. | Math., | press, | 2006. | | Earlier | version | available | at |

<http://www.i3s.unice.fr/fpsac/FPSAC03/articles.html>

10

[2] J. H. van Lint and R. M. Wilson, A Course in Combinatorics, 2nd ed., Cambridge University Press, NY, 2001.

[3] Neil J. Sloane (founder and maintainer), The On-Line Encyclopedia of Integer Se-quences [http://www.research.att.com:80/ njas/sequences/index.html?blank=1](http://www.research.att.com:80/~njas/sequences/index.html?blank=1)

11