

Bootstrap Percolation, Indecomposable Permutations, and the n -Kings Problem

Mark Huibregtse[†], Cristobal Lemus-Vidales[‡], and David Vella^{*}

[†] *Mathematics and Statistics Department, Skidmore College (Emeritus),
 9 Pinewood Ave., Saratoga Springs, NY, 12866,
 Email: mhuibreg@skidmore.edu*

[‡] *School of Computing and Data Science, Wentworth Institute of Technology,
 550 Huntington Ave., Boston, MA 02115,
 Email: lemusvidalesc@wit.edu*

^{*} *Mathematics and Statistics Department, Skidmore College (Emeritus),
 4087 Buck Creek Trl., Ballston Spa, NY, 12020,
 Email: dvella@skidmore.edu*

Received: August 9, 2025, **Accepted:** November 6, 2025, **Published:** November 21, 2025
 The authors: Released under the CC BY-ND license (International 4.0)

ABSTRACT: We study the process of bootstrap percolation on $n \times n$ permutation matrices, inspired by Shapiro and Stephens' article (SS): "Bootstrap percolation, the Schröder numbers, and the n -kings problem" (1991). In this percolation model, cells mutate (from 0 to 1) if at least two of their cardinal neighbors contain a 1, and thereafter remain unchanged; the process continues until no further mutations are possible. After carefully analyzing this process, we consider how it interacts with the notion of (in)decomposable permutations. We prove that the number of indecomposable permutations whose matrices "fill up" to contain all 1's (or are "full") is half of the total number of full permutations. This leads to a new proof of a key result in (SS), that the number of full $n \times n$ permutations is the $(n - 1)^{\text{st}}$ large Schröder number. Finally, after rigorously justifying a heuristic argument in (SS), we find a new formula for the number of $n \times n$ "no growth" permutation matrices, and hence a new solution to the well-known n -kings problem.

Keywords: Bootstrap percolation; Indecomposable permutation; n -kings problem; Schröder numbers
2020 Mathematics Subject Classification: 05A05; 05A15.

1. Introduction

1.1 Motivation

This article was inspired by a close reading of Shapiro and Stephens' paper [5], which studied the combinatorics of bootstrap percolation, a growth model in which the value of a cell in a 0-1 matrix can change from 0 to 1 whenever at least two of its neighbors (to the north, south, east, and/or west) have the value 1. After a cell has "mutated" from 0 to 1, its value remains constant thereafter. The process begins with an $n \times n$ permutation matrix, and proceeds in a series of steps consisting of the random selection of a "mutable" cell (if any) and its mutation; the process terminates when there are no more mutable cells.

Two possible outcomes of the process, illustrated by the following examples, are of particular interest:

Example 1.1. *The matrix "fills up" (each arrow represents two steps):*

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

The matrix is "no-growth":

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Indeed, Shapiro and Stephens set out to determine the number and asymptotic density of the $n \times n$ permutation matrices that 1) fill up, and 2) are no-growth. In the former case, using generating function techniques, they showed [5, Theorem 1, p. 276] that:

$$\begin{aligned} & \text{The number } p_n \text{ of } n \times n \text{ permutation matrices that fill up is } S_{n-1}, \\ & \text{the } (n-1)^{\text{st}} \text{ (large) Schröder number.} \end{aligned} \quad (1)$$

From this it follows that the asymptotic density $(S_{n-1})/n! \rightarrow 0$ as $n \rightarrow \infty$.

The (large) Schröder numbers: 1, 2, 6, 22, 90, 394, 1806, 8558, 41586, ... count, among other objects, the number of subdiagonal lattice paths from $(0,0)$ to (n,n) consisting of steps East $(1,0)$, North $(0,1)$ and Northeast $(1,1)$. For more information about this sequence, see the OEIS entry [6, A006318].

In the latter case, they obtained a fascinating answer in terms of a composition of generating functions. For $n \geq 0$, let a_n denote the number of $n \times n$ no-growth permutation matrices (with $a_0 = 1$), and let $A(x) = \sum_{n=0}^{\infty} a_n x^n$ be the generating function of this sequence. Let $R(x)$ denote the generating function for the Schröder numbers; hence $xR(x)$ is the generating function for the sequence $(S_{n-1})_{n=1}^{\infty}$. Then Shapiro and Stephens state that:

$$A(xR(x)) = \sum_{n=0}^{\infty} n! x^n = \varepsilon(x), \quad (2)$$

justifying the result with a very brief intuitive summary. They then note that the functional inverse of $xR(x)$ is $x((1-x)/(1+x))$, yielding the following expression for $A(x)$:

$$A(x) = \varepsilon(x((1-x)/(1+x))), \quad (3)$$

which, as they said, “at least in some sense,” provides a computation of a_n and permits the asymptotic density of the no-growth permutation matrices to be deduced. It was this formula that particularly sparked our interest in [5], since a result of D. Vella [7] on composite generating functions would permit the extraction of an explicit formula for the no-growth numbers: 1, 1, 0, 0, 2, 14, 90, 646, 5242, ... The sequence (a_n) appears in the OEIS as entry [6, A002464].

Before writing up this new explicit formula (it appears below in the concluding Section 6), we wanted to make sure we thoroughly understood the proof of (2). This spurred us to undertake a close reading of Shapiro and Stephens’s paper.

The next section discusses the new results we obtained and lays out the overall structure of this paper. Before beginning this survey, we note a few conventions that are used throughout the paper:

- The set $\{1, 2, \dots, n\}$ of positive integers less than or equal to n is denoted $[n]$.
- A permutation $\pi = a_1 a_2 \dots a_n$ of $[n]$ is written in **single-line notation**, that is, given $\pi : [n] \rightarrow [n]$, we have $a_i = \pi(i)$ for $1 \leq i \leq n$.

1.2 Results and Organization

In Section 2, we discuss the process of bootstrap percolation on permutation matrices in detail. We establish some elementary but useful results, including Theorem 2.1, which states that the final result of percolation does not depend on the order in which mutable cells are changed from 0 to 1, and Theorem 2.2 and its corollary, which precisely describe the possible final configurations of percolation on a permutation matrix.

We then turn in §2.4 to a study of “tile merging” algorithms for percolation. The first part of Example 1.1 gives the idea: We first merge two 1×1 tiles (the $(2,1)$ and $(3,2)$ cells) to obtain a 2×2 tile, which is then merged with the 1×1 tile (at $(1,3)$) to yield the final 3×3 tile. Looking at each tile merge as a single step, one achieves greater efficiency of execution and conceptual clarity.

Tile merging is not mentioned explicitly in [5], but is implicit in the corresponding idea of “bracketing” of permutations, introduced in [5, §2] and here discussed in §2.5. The basic idea of bracketing is to encode the sub-tiles that are created during a tile merging sequence with subsequences of the permutation: each sub-tile T corresponds to the initial 1’s in the permutation matrix that belong to T . In the preceding example, the permutation is $\pi = 213$, and the result of what we call “left-bracketing” is $([21]3)$. This records that the first tile merge joined the 1×1 tiles representing the 1’s corresponding to $a_1 = 2$ and $a_2 = 1$, and the second joined the new 2×2 tile with the 1×1 tile representing the 1 corresponding to $a_3 = 3$. We use parens (m_1, m_2)

(resp. square brackets $[m_1, m_2]$) to denote that the components in m_1 are all less than (resp. greater than) the components in m_2 .

If a permutation matrix associated to a permutation π fills up under bootstrap percolation, or is “full,” as we shall say, then tile merging/bracketing must lead to a final $n \times n$ tile or single “top-level bracketing” having one of the forms (m_1, m_2) or $[m_1, m_2]$; this is discussed in §2.6. This bracketing carries valuable information regarding the underlying permutation, as will be seen.

It turns out to be important to be precise in describing an algorithm for tile merging or (equivalently) bracketing. The semi-precise bracketing process discussed in [5, p. 277] is in some cases compatible with multiple interpretations that can lead to errors if the wrong interpretation is used; we exemplify this in §2.7, where we also state and prove a fully valid version of the potentially erroneous assertion in [5] that could result from such a misinterpretation.

In §3, we begin consideration of an issue not discussed in [5], namely, the interplay of the concepts of (in)decomposable permutation and bootstrap percolation. §3.1 recalls the definitions, and §3.2 discusses the indecomposable components of a given permutation and how to find them. This discussion is extended in §4, where we consider (in)decomposable permutations that are full. In §4.1, we show that the operation of inversion on permutations ($a_1 a_2 \dots a_n \mapsto a_n a_{n-1} \dots a_1$) carries full permutations to full permutations; we also show that a full indecomposable (resp. decomposable) permutation has top-level bracketing (under any bracketing algorithm) of the form $[m_1, m_2]$ (resp. (m_1, m_2)). Consequently, since (as we show) inversion changes the form of the top-level bracketing of a full permutation ($[,]$ to $(,)$ and vice versa), in §4.2 we obtain a key result (Corollary 4.1, the “Half-lemma”): for $n \geq 2$, the number of full indecomposable permutations of $[n]$ is exactly half of the number of full permutations.

We then consider the indecomposable components of a full permutation in §4.3. Theorem 4.2 shows how the rightmost indecomposable component of a full permutation can be computed through (left) bracketing; iteration (Example 4.2) then leads to a process for extracting a permutation’s full indecomposable factorization. This section concludes with Theorem 4.3, which establishes that a permutation π is full if and only if each of its indecomposable components is full, a result needed in the sequel.

In §5, we offer a new proof of Shapiro and Stephens’ result (1) that the number p_n of full permutations of $[n]$ equals the $(n - 1)^{\text{st}}$ (large) Schröder number. Instead of using generating function techniques, our proof shows (using the Half-lemma) that the sequence $(p_n)_{n \geq 1}$ satisfies the same recurrence as does the Schröder sequence.

Finally, to bring the paper to a close, we offer a detailed proof of equation (2), and then show how it leads to an explicit formula for the number a_n of no-growth permutations of $[n]$. Since the $n \times n$ no-growth permutation matrices correspond to the arrangements of n non-attacking kings on the $n \times n$ chessboard, we thereby attain an explicit solution to the well-known n -kings problem.

2. Bootstrap Percolation

This section studies the process of bootstrap percolation on permutation matrices. It includes a proof that the final configuration attained does not depend on the order in which “mutable” cells are changed from 0 to 1, and a discussion, from our point of view, of the bracketing process (an efficient means for determining the final configuration) introduced in [5].

2.1 The process of bootstrap percolation

Let M be an $n \times n$ permutation matrix (a matrix whose entries are 0’s and 1’s and in which every row and every column contains exactly one 1). Bootstrap percolation is an iterative process in which certain “mutable” entries (or cells) of M are changed from 0 to 1.

Definition 2.1. A cell of M is **mutable** if it contains a 0 and at least two of its neighbors (to the north, south, east, and/or west) contain a 1; when the cell is changed, we say it has **mutated**, and is therefore no longer mutable.

Each step in bootstrap percolation consists of choosing a mutable cell (if any) in the current configuration of the matrix and mutating it. If several mutable cells are available at any stage, the choice of which to select and mutate is arbitrary. Percolation continues until there are no remaining mutable cells, at which point it terminates in a final configuration. The following example illustrates the process:

Example 2.1.

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Remark 2.1. The preceding example illustrates our convention for associating a permutation with a permutation matrix. We view the initial permutation matrix as the “graph” of the permutation 34152, that is,

$$1 \mapsto 3, 2 \mapsto 4, 3 \mapsto 1, 4 \mapsto 5, \text{ and } 5 \mapsto 2.$$

In the next two sections, we show that the final result attained by bootstrap percolation is independent of the order in which mutable cells are mutated, and we describe the properties of the final configuration.

2.2 Invariance of final result of bootstrap percolation

The purpose of this section is to prove that the final configuration of bootstrap percolation is independent of the order in which the percolation is carried out (i.e., the order in which mutable cells are mutated).

Given an $n \times n$ permutation matrix M , there are n cells containing a 1 and $n^2 - n$ cells containing a 0. It follows that bootstrap percolation will require at most $n^2 - n$ steps to reach a terminal position.

Definition 2.2. Given a cell $c = c_{i,j}$ in M , we say that c is **potentially mutable** if c initially contains a 0 and there is a sequence of percolation steps that mutates it. We then define

$$L(c) = \begin{cases} \text{the minimum number of steps required to mutate } c, \text{ if } c \text{ is} \\ \text{potentially mutable, and otherwise} \\ n^2 \text{ (which exceeds the maximum possible number of steps).} \end{cases}$$

We say that a percolation sequence is **complete** if it ends in a position with no mutable cells, and therefore no further steps are possible. Finally, we define a collection of subsets of cells in M as follows:

$$\begin{aligned} U_0 &= \{c : c \text{ is initially set to } 1\}, \text{ and} \\ U_i &= \{c : L(c) = i\}, \text{ for } 1 \leq i \leq n^2 - n. \end{aligned}$$

Clearly $|U_0| = n$; note it is possible for all the U_i for $1 \leq i \leq n^2 - n$ to be empty, which occurs if the original permutation matrix M is in a no-growth configuration (meaning no cell in M is mutable, or, equivalently, $U_1 = \emptyset$).

Example 2.2. Consider the following 3×3 permutation matrix representing the permutation $\pi = 213$:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

In this matrix, one sees by inspection that

$$\begin{aligned} U_1 &= \{c_{\{2,2\}}, c_{\{3,1\}}\}, \quad U_2 = \{c_{\{1,2\}}, c_{\{2,3\}}\}, \quad U_3 = \{c_{\{1,1\}}, c_{\{3,3\}}\}, \text{ and} \\ U_4 &= \emptyset, \quad U_5 = \emptyset, \text{ and } U_6 = U_{3^2-3} = \emptyset. \end{aligned}$$

We now turn to the invariance proof, for which we need the following Lemma:

Lemma 2.1. If a cell c containing a 0 is mutable before a percolation step is carried out, and that step does not change its value, it remains mutable after the step is carried out.

Proof. The number of neighbors of c containing a 1 either stays the same or increases. □

Theorem 2.1. Any complete sequence of percolation steps mutates the same set of cells as any other, which implies that the final configuration attained does not depend on the order in which the percolation steps are carried out.

Proof. We will show that any complete sequence of percolation steps has a final configuration in which the set of cells containing a 1 is equal to $\bigcup_{i=0}^{n^2-n} U_i$. The desired result is then immediate.

In case the permutation matrix M is no-growth, all the $U_i = \emptyset$ except for U_0 , which is the set of cells containing the 1's of M . In this case, there is only one complete sequence of percolation steps, namely, the sequence of no steps at all, so the final configuration is indeed given by $\bigcup_{i=0}^{n^2-n} U_i$ in this case.

If we are not in the no-growth case, we must have that $U_1 \neq \emptyset$, and all the cells $c \in U_1$ are mutable. The first percolation step must be to mutate one of the cells in U_1 . By the Lemma, the remaining cells in U_1 remain mutable, and will continue to remain so (if left unmutated) as percolation continues. Since a complete percolation sequence must continue until there are no more mutable cells, eventually all the cells in U_1 will be mutated. At that point, every element in U_2 will either contain a 1 from a previous step or be mutable, since every cell mutable in one step has been mutated. By the same argument, eventually all of the cells in U_2 will be mutated, at which point any unmutated cells in U_3 will be mutable. Continuing in this way, we see that $\bigcup_{i=0}^{n^2-n} U_i$ is a subset of the final configuration of 1's, but since the maximum number of steps is equal to $n^2 - n$, we know that there cannot be any other 1's in the final configuration. This completes the proof. \square

Remark 2.2. *Theorem 2.1 is an easy consequence of Newman's Lemma [4] (see [10] for a nice discussion); we thank Ira Gessel for this observation.*

2.3 The final configuration of bootstrap percolation

In this section, we study the final configuration of bootstrap percolation, that is, the pattern of 1's in the matrix at the end of a complete percolation sequence. Here and elsewhere, the following terminology is useful:

Definition 2.3. *Given a matrix $(c_{i,j})$, a $(k \times l)$ **tile** is a rectangular sub-region of the matrix containing the following cells:*

$$\{c_{i,j} : i_0 \leq i \leq i_0 + k - 1 \text{ and } j_0 \leq j \leq j_0 + l - 1\}.$$

*If every cell of a tile T contains a 0 (resp. 1), we call T a **null** (resp. **unitary**) tile (in particular, an individual matrix entry (a 1×1 tile) is null (resp. unitary) if it contains a 0 (resp. 1). Lastly, if T is a $k \times k$ tile, we say T is a **square** tile of **size** k .*

The main result of this section can then be summarized as follows: the final configuration of bootstrap percolation on an $n \times n$ permutation matrix consists of m (for some $1 \leq m \leq n$) square unitary tiles in a no-growth configuration.

Imagine the percolation process broken down into steps:

$$M_0 \longrightarrow M_1 \longrightarrow M_2 \longrightarrow \dots \longrightarrow M_r, \quad (4)$$

where M_0 is an $n \times n$ permutation matrix, each step involves mutating a single cell, and M_r is the final matrix after percolation ends. We have the following lemma:

Lemma 2.2. *Every row and every column in each M_t contains exactly one unbroken string of unitary entries. In other words, a row or a column cannot contain any null entries between any two unitary entries, and no row or column can be null (i.e., consist entirely of null entries).*

Proof. We induct on t . The result holds when $t = 0$ because we are starting with a permutation matrix M_0 , which has exactly one unitary entry in each row and column. Assume the result holds for some M_t , $1 \leq t < r$. The only difference between M_t and M_{t+1} is that a single cell, say $c = c_{i,j}$, has mutated, meaning that c is mutable in M_t . This means that c is null in M_t and that at least two unitary entries in M_t are adjacent to c . If the latter two entries were in the same row (or the same column) as c , with c adjacent to them both, then c must be between them, which would violate the inductive hypothesis (M_t would contain a row or a column with unitary entries separated by a null entry). It follows that those unitary entries that are adjacent to c must be in different rows and columns, but since they are both adjacent to c , they must be diagonally adjacent to each other, one in the same (i^{th}) row as c , and the other in the same (j^{th}) column. Then, when c is mutated, the i^{th} row gains a unitary entry that is adjacent to an existing unitary entry in M_t already in the i^{th} row, and similarly, the j^{th} column gains a unitary entry adjacent to a unitary entry of M_t already in the j^{th} column. Thus each row and column in M_{t+1} is the same as that in M_t , except for the i^{th} row and the j^{th} column, which have each acquired a new unitary entry adjacent to a previous unitary entry in that row or column. Therefore, no gaps are produced, and the unitary entries in each row and column continue to form a single unbroken string. So the Lemma holds for M_{t+1} , and so for all M_t by induction. \square

Applying Lemma 2.2 to the final matrix M_r leads to the following Theorem:

Theorem 2.2. *The final matrix M_r consists of square unitary tiles that are separated by null entries. Furthermore, each row and column of M_r meets exactly one of these unitary tiles.*

Proof. Consider the final matrix M_r . We know the first column has a string of consecutive unitary entries, and so does the second column. Since M_r is the final matrix, the percolation process has terminated. So, the unitary strings in the first two columns either occupy the exact same rows, or else they are separated by enough null cells to prevent further percolation. Any other configuration—such as being diagonally adjacent, or partially overlapping—would lead to further percolation, a contradiction. In case the strings in the first two columns do match rows exactly, look at the unbroken string in the third column. It must occupy the exact same rows as the string in the second, or be separated by enough zeros to avoid percolation, etc. The upshot is, by iteration, for several columns, all the unbroken strings are in the exact same rows. The first time one hits a column, say the $(k+1)^{\text{st}}$, where the unbroken string is not in those rows, the remaining cells in those rows in M_r are all null, otherwise, we would contradict Lemma 2.2. That means that the unitary entries in the first k columns form an unbroken rectangular unitary tile. Note that Lemma 2.2 implies that both the rows and columns passing through this tile meet no other unitary entries outside of this tile.

At this point, if we haven't filled up the matrix completely, and there is a $(k+1)^{\text{st}}$ column with its unbroken string of unitary tiles in a different set of rows than the first tile, we iterate this process, producing a second rectangular unitary tile, in consecutive columns beginning with the $(k+1)^{\text{st}}$, with all the unitary entries in the same set of rows. By iteration, this shows M_r consists of rectangular unitary tiles, separated by null cells, and the additional fact that each row and column of M_r meets exactly one of these unitary tiles (because of Lemma 2.2).

It remains to show that these unitary tiles are actually square. Consider one of them and suppose it has size $p \times q$. Looking at the q columns making up this tile, we observe that in the initial matrix M_0 , each of those columns has a unitary entry which survives the percolation process and ends up in M_r , so *must* be within the rows meeting this tile in M_r , since the columns meeting this tile meet no other unitary entries outside the tile. However, those original unitary entries from M_0 must all be in distinct rows since M_0 is a permutation matrix. It follows that $p \geq q$. By making the same argument working with rows (or applying the same argument to the transpose matrix), we get that the p unitary entries from those rows in M_0 must end up in (distinct) columns inside the tile, so $q \geq p$. It follows that $p = q$, so the unitary tiles are, in fact, square, completing the proof of the Theorem. \square

Since each column of M_r meets exactly one of the m square unitary tiles, we can unambiguously order these tiles from left to right, and label them T_1, T_2, \dots, T_m . We denote the size of T_j by s_j ; it is then a clear consequence of Lemma 2.2 that $\sum_{j=1}^m s_j = n$ (the dimension of the matrices M_0, M_1, \dots, M_r).

To complete the work of this section, we show that, as asserted above, the unitary tiles T_j of M_r are in a no-growth configuration. The idea is to condense the matrix M_r into an $m \times m$ permutation matrix in which (reading left-to-right) the j^{th} unitary cell corresponds to the j^{th} unitary tile T_j of M_r . Intuitively, one does this by “viewing the matrix M_r from far away,” so that the T_j appear to be single cells (or size 1 square unitary tiles). More precisely, one can construct an $m \times m$ matrix by crossing out, for each $1 \leq j \leq m$, $s_j - 1$ of the rows and $s_j - 1$ of the columns of M_r that meet T_j . What remains is an $m \times m$ permutation matrix CM (because each row and each column of CM contains exactly one unitary cell and $m - 1$ null cells) that is a “condensed” version of M_r : the j^{th} unitary cell in CM (from the left) corresponds to the tile T_j in M_r . We now have the following.

Corollary 2.1. *The $m \times m$ permutation matrix CM is no-growth, that is, it contains no mutable cells.*

Proof. Suppose to the contrary that there is a mutable cell $c_{i,j}$ in CM . Then $c_{i,j}$ is null and has a unitary cell as either west or east neighbor and a unitary cell as either north or south neighbor, such that the latter two cells are diagonally adjacent. Consider the case where $c_{i,j-1} = 1$ is the west neighbor and $c_{i+1,j} = 1$ is the south neighbor of $c_{i,j}$. Then we know that the west neighbor corresponds to the unitary tile T_{j-1} and the south neighbor to the unitary tile T_j . Since M_r is the final configuration, we know that T_{j-1} and T_j cannot meet diagonally at a corner, otherwise there would be additional percolation steps to perform. Accordingly, since the rightmost column of T_{j-1} is adjacent to the leftmost column of T_j , there must be one or more rows of M_r that separate the southernmost row of T_{j-1} and the northernmost row of T_j . Consider one of these separating rows, which must intersect a unitary tile T_k distinct from T_{j-1} and T_j . After crossing out $s_k - 1$ of the s_k rows meeting T_k (all of which are separating rows), the remaining row intersecting T_k must condense to a row in

CM that vertically separates the cells $c_{i,j-1}$ and $c_{i+1,j}$, which is a contradiction. Similar contradictions obtain in the other cases, completing the proof. \square

The following definition highlights a particular case of interest:

Definition 2.4. We say that a permutation π of $[n]$ is **full** if the final configuration of bootstrap percolation on the permutation matrix associated to π consists of a single unitary tile of size n .

2.4 Tile merging algorithms for percolation

Let $\pi = a_1 a_2 \dots a_n$ be a permutation of $[n]$ and let M_0 denote the permutation matrix corresponding to π . By Theorem 2.1, we know the final configuration M_r of bootstrap percolation on M_0 is independent of the sequence in which mutable cells are mutated. In this section, we discuss algorithms for bootstrap percolation that provide conceptual insight and allow for rapid computation of the final configuration M_r . The basic idea is to replace the mutation of individual cells with the “merging” of diagonally-adjacent square unitary tiles. We will soon see how each tile-merging algorithm corresponds to a bracketing process on the string “ $a_1 a_2 \dots a_n$.”

Example 2.3. Here is an example of the merging of two diagonally-adjacent square unitary tiles of sizes 2 and 1. Although the merging requires four mutation steps, we conceive of the merging as a single “macro” step, and don’t specify the order in which the mutations will occur. Also note that if two square unitary tiles of sizes p and q are merged, the result is a square unitary tile of size $(p + q)$.

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

In brief, a **tile merging algorithm** proceeds as follows: One begins with the $n \times n$ permutation matrix M_0 , which contains n square unitary tiles of size 1. One repeatedly searches the square unitary tiles in the matrix (in some specified order), and merges diagonally-adjacent pairs until no more merges are possible, at which point the algorithm terminates.

On termination, there are two possibilities:

- i. There is a single square unitary tile of size n . This happens when the original permutation is full.
- ii. There are m , for $1 < m \leq n$, square unitary tiles in M_r that are in a no-growth configuration, since no further merges are possible.

It follows that the final configuration has the form described in Section 2.3, that is, it will be a matrix containing $1 \leq m \leq n$ square unitary tiles in a no-growth configuration, which implies by definition that a complete sequence of cell mutations has been carried out. Consequently, by Theorem 2.1, any other complete sequence of cell mutations beginning with the original permutation matrix will lead to the same final configuration.

We are primarily concerned with the algorithms named in the following definition:

Definition 2.5. The **left merging algorithm** proceeds as follows: The matrix being percolated (starting with M_0) is scanned from left to right in search of a diagonally adjacent pair of square unitary tiles. If such a pair is found, the tiles are merged and the process is repeated until a traversal finds no diagonally adjacent pair to merge, at which point the process terminates. It is clear that termination must occur, since the number of square unitary tiles decreases by 1 each time a merge occurs. The **right merging algorithm** is essentially the same, except that the matrix being percolated is repeatedly scanned from right to left.

Here is an example of these algorithms on a 4×4 (full) permutation matrix; each arrow represents the execution of one traversal and merge operation:

Example 2.4. Let $\pi = (1324)$, a (full) permutation of $[4]$. We see that left merging and right merging lead to the same final configuration (as they must), but through different pathways.

Left merging:

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Right merging:

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

2.5 Bracketing of permutations

Given a permutation $\pi = a_1 a_2 \dots a_n$ of $[n]$, Shapiro and Stephens, in [5], show how to determine if π is full, or more generally, to determine the final configuration of bootstrap percolation, by bracketing the string $a_1 a_2 \dots a_n$ while avoiding the need to compute the sequence of matrices (4). In effect, the bracketing keeps track of the progress of a tile merging algorithm by encoding tiles as substrings of $a_1 a_2 \dots a_n$. The key idea is given in Lemma 2.3, for which the following definition is useful:

Definition 2.6. Let π be a permutation of $[n]$, let M_0 be the permutation matrix of π , and let T_0 be a square tile of size k ($1 \leq k \leq n$) in M_0 . Consider a complete sequence of percolation steps as in (4). For each $0 < t \leq r$, let T_t be the tile in M_t occupying the same positions as T_0 occupies in M_0 . We call T_t the t^{th} **descendant** of T_0 and T_0 the **origin** of T_t . Note that if T_0 contains k non-zero entries of M_0 , then T_0 can be viewed as a permutation matrix for a permutation of $[k]$, since each row and column of M_0 (and hence T_0) contains only one non-zero entry. If in addition T_0 represents a full permutation of k , then for some t we will have that T_t is a unitary square tile of size k ; in this case we call T_t a **good** tile.

Example 2.5. The following figure shows the complete percolation of a 7×7 permutation matrix. The final configuration (to the right) consists of five square unitary tiles, four of size 1 and one of size 3. All five of these tiles are good (in agreement with the following Lemma), but the 4×4 tile in the northwest, comprising the cells $c_{i,j}$ such that $1 \leq i, j \leq 4$, although it represents a permutation of $[4]$, is not a good tile, since the permutation it represents is not full.

$$\begin{pmatrix}
 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}
 \xrightarrow{6 \text{ steps}}
 \begin{pmatrix}
 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1
 \end{pmatrix}$$

Lemma 2.3. Let $\pi = a_1 a_2 \dots a_n$ be a permutation of $[n]$ and let M_0 be the permutation matrix of π . If a tile merging algorithm is run on M_0 , then every square unitary tile T that appears during the computation is a good tile.

Proof. The Lemma is clearly valid for the n square unitary tiles of size 1 that are the unique unitary cells of M_0 . Arguing by contradiction, we suppose the Lemma is false and consider a square unitary tile T of minimum size $k > 1$ such that: 1) T results from the merging of two previously-appearing square unitary tiles T' and T'' , and 2) T is not a good tile. By the minimality of k , we know that T' and T'' have sizes $k' < k$ and $k'' < k$, respectively, hence are good tiles. Consequently, their origins T'_0 and T''_0 contain k' and k'' unitary entries, respectively, so the origin T_0 of T contains $k' + k'' = k$ unitary entries and therefore represents a permutation of $[k]$. Moreover, since T is obtained from the merging of T' and T'' , it is clear that T_0 represents a full permutation; consequently, T is a good tile, contradicting the hypothesis and completing the proof of the Lemma. \square

Remark 2.3. Lemma 2.3 implies that every square unitary tile T of size k that appears during the execution of a tile merging algorithm corresponds uniquely to a length- k substring “ $a_j a_{j+1} \dots a_{j+k-1}$ ” that encodes the list of unitary cells in T_0 . For instance, in Example 2.4, the tile of size 2 that results from the first step of both left and right merging corresponds to the substring “32” of $\pi = 1324$. However, the result of the second step for left merging differs from the result for right merging: for left merging, the substring is “132,” and for right merging, it is “324.” This is the key idea behind the bracketing process. We will call the (suitably bracketed) substrings representing the good tiles “melds” (they are called “blocks” in [5]). A formal definition follows:

Definition 2.7. Let $\pi = a_1 a_2 \dots a_n$ be a permutation of $[n]$. We inductively define a **meld** of π as follows:

- i. Every component a_j of π is a meld, and
- ii. If m_1 and m_2 are consecutive melds such that the union of the components of m_1 and m_2 is a (permuted) set of consecutive integers in $[n]^*$, then (m_1, m_2) (resp. $[m_1, m_2]$) is the meld resulting from their merger in case $\max(m_1) = \min(m_2) - 1$ (resp. $\min(m_1) = \max(m_2) + 1$).

*This condition is equivalent to the tiles corresponding to m_1 and m_2 being diagonally adjacent.

To complete the bracketing of π , one mimics the tile merging algorithm being used, but instead of merging square unitary tiles in the percolating matrix, one merges the melds that encode those tiles. To illustrate this, we mimic the computations in Example 2.4 to compute the left and right bracketings of $\pi = 1324$:

$$\begin{aligned} \text{Left bracketing: } 1\ 3\ 2\ 4 &\mapsto 1\ [3\ 2]\ 4 \mapsto (1\ [3\ 2])\ 4 \mapsto ((1\ [3\ 2])\ 4), \\ \text{Right bracketing: } 1\ 3\ 2\ 4 &\mapsto 1\ [3\ 2]\ 4 \mapsto 1\ ([3\ 2]\ 4) \mapsto (1\ ([3\ 2]\ 4)). \end{aligned}$$

In this case, since the bracketings terminate with a single meld encompassing all four digits in π , we confirm our earlier observation that π is full.

2.6 “Top-level bracketings” of a full permutation

Lemma 2.4. *Let π be a full permutation of $[n]$, for $n > 1$. Then, for any bracketing algorithm, π has a “top-level” bracketing of the form (m_1, m_2) or $[m_1, m_2]$ that comprises the integers in $[n]$.*

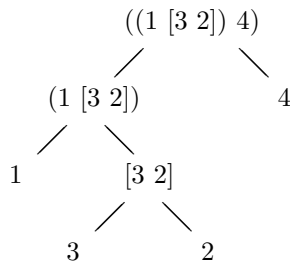
Proof. The hypothesis implies that the bracketing process on π must terminate with a single meld comprising the integers in $[n]$. The initial number of melds is $n \geq 2$, corresponding to the n digits in π , and this number decreases by 1 every time two melds are merged. Since the process terminates with a single meld that must arise from the merging of two melds m_1 and m_2 , the result follows. \square

Remark 2.4. *Given our convention on how a permutation π of $[n]$ is mapped to its permutation matrix (see Remark 2.1), one checks that the orientation of the tiles T_1 and T_2 corresponding to the melds m_1 and m_2 in the top-level bracketing have the following orientations prior to their merger:*

$$\begin{aligned} (m_1, m_2) &\leftrightarrow \begin{pmatrix} 0 & T_2 \\ T_1 & 0 \end{pmatrix}, \\ [m_1, m_2] &\leftrightarrow \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix}. \end{aligned}$$

2.7 The binary tree associated to the bracketing of a full permutation

As explained in [5, p. 277], a bracketing of a full permutation can be displayed as a binary tree in which the vertices are the melds, and the left (resp. right) child of a meld of the form (m_1, m_2) or $[m_1, m_2]$ is m_1 (resp. m_2), and the singleton melds are the leaves of the tree. For example, the following figure shows the tree associated to the full (left-bracketed) permutation $\pi = ((1[32])4)$:



Lemma 2.5. *Let $n > 1$ and let π be a permutation of $[n]$. If left bracketing is performed on π , the corresponding binary tree has the following property: the right child m_2 of a non-leaf node (m_1, m_2) (resp. $[m_1, m_2]$) is either a singleton or has the form $[m_{21}, m_{22}]$ (resp. (m_{21}, m_{22})); in brief, the type of the right child of a non-leaf node under left bracketing differs from the type of its parent. Symmetrically, the left child of a non-leaf node under right bracketing differs from the type of its parent.*

Proof. Suppose to the contrary that under left bracketing, a non-leaf node has the form (m_1, m_2) , where $m_2 = (m_{21}, m_{22})$. Recall from Remark 2.3 that each of the melds m_1 , m_{21} , m_{22} , and (m_1, m_2) comprises a (permuted) consecutive run of integers in $[n]$, and the bracketing indicates that the integers in m_1 are all less than the integers in m_2 and that the integers in m_{21} are all less than those in m_{22} . A moment’s reflection then implies that (under left merging) m_1 and m_{21} will be merged to form (m_1, m_{21}) before m_{21} and m_{22} can be merged, contradicting the hypothesis. The proof in the case of a supposed non-leaf node of the form $[m_1, m_2]$ with $m_2 = [m_{21}, m_{22}]$ is similar, as is the proof of the last statement in the case of right bracketing. \square

Remark 2.5. The left-bracketing version of the assertion in Lemma 2.5 is stated at the top of page 278 of [5]; subsequently, it is needed there to establish the count of full permutations of $[n]$ for $n \geq 2$. However, this result need not be true depending on one's interpretation of the (somewhat vague) bracketing procedure given on page 277 of [5]:

The permutation π can be considered as a sequence of [melds].
We read π left to right and use [the merging rule (ii) of Definition 2.7)]
to form new [melds] whenever possible. (5)

The following example makes the key issue clear:

Example 2.6. Consider bracketing the permutation $\pi = 4231$. Beginning at the left, we find that the first adjacent pair of melds that can be merged is $(2, 3)$. Merging them yields $4(23)1$, leaving two adjacent pairs of mergeable melds $(4, (23))$ and $((23), 1)$. There are now two possible ways to complete the bracketing:

$$\begin{aligned} \text{i. } 4(23)1 &\mapsto [4(23)]1 \mapsto [[4(23)]1], \\ \text{ii. } 4(23)1 &\mapsto 4[(23)1] \mapsto [4[(23)1]]. \end{aligned}$$

Both computations are compatible with the procedure sketched in (5), but only the first (the result of left merging) satisfies the left-bracketing version of Lemma 2.5. The second computation (in which newly-merged melds are, if possible, merged with their right-hand neighbors as a left-to-right traversal of π continues) appears, based on the example given on page 277 of [5], to be what was intended by the procedure (5).

In the sequel, we will have occasion to use the left-bracketing version of Lemma 2.5, and will only do so when left bracketing has been used.

3. Indecomposable permutations

In this section, we recall the definition and basic theory of indecomposable permutations. The concept is due to Comtet in the early 1970's (See [2] or [3]). Indecomposable permutations are sometimes known as *irreducible* or *connected* permutations.

3.1 Definition of indecomposable permutation

Definition 3.1. A permutation π of $[n]$ is called **indecomposable** if there does not exist an index $1 \leq k < n$ such that $\pi(\{1, 2, \dots, k\}) = \{1, 2, \dots, k\}$. That is, there does not exist a proper initial segment of $\{1, 2, \dots, n\}$ that is mapped onto itself by σ ; equivalently, no length- k prefix of the single-line representation of π is a permutation of $[k]$ for some $k < n$. Otherwise, we say π is **decomposable**.

Example 3.1. The permutation $\tau = 4132675$ is decomposable because 4132 is a permutation of $\{1, 2, 3, 4\}$, while the permutation $\sigma = 4167523$ is indecomposable.

Suppose $\pi = a_1 a_2 \dots a_n$ is a permutation of a set of integers (or any linearly ordered set). Following [3], we define the **reduced form** of π to be the permutation of $[n]$ obtained by replacing the i^{th} smallest element of π with i . We denote the reduced form of π by π_{red} . For example, if $\pi = 2754$ (a permutation of $\{2, 4, 5, 7\}$), then $\pi_{\text{red}} = 1432$. This notion permits us to extend the definition of indecomposable permutation to permutations of arbitrary (finite) linearly ordered sets: given any such partition π , we say that π is **indecomposable** provided that π_{red} is indecomposable.

Example 3.2. $\pi = 2754$ is decomposable because its reduced form 1432 is decomposable.

3.2 Indecomposable components of a permutation

Let $\pi = a_1 a_2 \dots a_n$ be a permutation of a linearly ordered set, and let $\pi_{\text{red}} = a'_1 a'_2 \dots a'_n$. Then let k denote the least positive integer such that $a'_1 a'_2 \dots a'_k$ is a permutation of $[k]$, and let $\pi_1 = a_1 a_2 \dots a_k$. One then checks easily that:

- i. $(\pi_1)_{\text{red}} = a'_1 a'_2 \dots a'_k$,
- ii. $a'_1 a'_2 \dots a'_k$ and (hence) π_1 are indecomposable,
- iii. π_1 is the maximum-length indecomposable prefix of π , and
- iv. π is indecomposable $\Leftrightarrow \pi = \pi_1$.

It is now easy to see that any permutation π can be written as a product (*i.e.*, concatenation) of one or more indecomposable factors, as follows:

$$\pi = \pi_1 \pi_2 \dots \pi_q,$$

where each π_j is the maximum-length indecomposable prefix of $\pi_j \pi_{j+1} \dots \pi_q$, for $1 \leq j \leq q$. The factors π_j are called the **indecomposable components** of π . For example:

Example 3.3. Let $\text{mip}(\pi)$ denote the *maximum-length indecomposable prefix* of the permutation π . Then the indecomposable components of $\pi = 241358679$ can be computed as follows:

$$\begin{aligned} \pi_1 &= \text{mip}(241358679) = 2413, \\ \pi_2 &= \text{mip}(58679) = 5, \\ \pi_3 &= \text{mip}(8679) = 867, \\ \pi_4 &= \text{mip}(9) = 9. \end{aligned}$$

In section 4.3, we show how the indecomposable components of a full permutation can be computed by using left bracketing.

4. Full Indecomposable Permutations and the “Half-lemma”

The main goal of this section is to prove that exactly half of the full permutations of $[n]$, for $n \geq 2$, are indecomposable. We also show that the indecomposable components of a full permutation can be computed by left bracketing, and prove that a permutation is full if and only if its indecomposable components are full.

4.1 An involution on full permutations

Given a permutation $\pi = a_1 a_2 \dots a_n$, we define the **reversal** of π to be the permutation $\pi^R = a_n a_{n-1} \dots a_1$. It is clear that the reversal map $R : \pi \mapsto \pi^R$ is an involution on the set of all permutations. Furthermore, it is easy to see that R is an involution on the set of full permutations:

Lemma 4.1. *If π is a full permutation of $[n]$, then so is π^R , and conversely.*

Proof. By Theorem 2.1, the final configuration resulting from bootstrap percolation is independent of the order in which cells are mutated. That being so, it suffices to note that running the left merging algorithm on π^R is the “mirror image” of running the right merging algorithm on π , so if the latter’s final configuration is a square unitary tile of size n , so is the former’s, and vice versa. \square

To clarify the preceding proof, we offer the following

Example 4.1. Let $\pi = 31254$ and $\pi^R = 45213$. We now perform the first three steps of right merging on π and left merging on π^R . After each step, the results are mirror images of one another. In this case, both π and π^R are full, since the unitary tiles of sizes 2 and 3 in the rightmost matrices will merge to yield unitary tiles of size 5.

Right merging on π :

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Left merging on π^R :

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Our next goal is Theorem 4.1, which states that the restriction of the involution R to full permutations sends indecomposable permutations to decomposable permutations and vice versa. For the proof, we need the following lemma:

Lemma 4.2. *If a permutation π of $[n]$, for $n \geq 2$, is full, then π is indecomposable if and only if the top-level bracketing (see Lemma 2.4) of π has the form $[m_1, m_2]$. Equivalently, π is decomposable if and only if the top-level bracketing of π has the form (m_1, m_2) .*

Proof. It suffices to prove one of the equivalent statements. We prove the second.

(\Rightarrow) Let π be a full decomposable permutation of $[n]$, for $n \geq 2$, and let m_1 and m_2 be the melds in the top-level bracketing of π . By definition of decomposable, we know that π maps $[k]$ to $[k]$ for some $1 \leq k < n$. We must show that the top-level bracketing has the form (m_1, m_2) . Suppose to the contrary that the top-level bracketing of π has the form $[m_1, m_2]$ where m_1 is a meld of length $p < n$ comprising the p largest values in $[n]$; that is, π maps $[p]$ to $\{n - p + 1, n - p + 2, \dots, n\}$. We now have the following cases:

Case 1: $k \leq p$. It follows that

$$\{1, 2, \dots, k\} \subseteq \{n - p + 1, n - p + 2, \dots, n\} \Rightarrow 1 = n - p + 1 \Rightarrow p = n, \text{ contradiction.}$$

Case 2: $k > p$. It follows that

$$\{n - p + 1, n - p + 2, \dots, n\} \subseteq \{1, 2, \dots, k\} \Rightarrow k = n, \text{ contradiction.}$$

Since both cases lead to contradictions, we have that the top-level bracketing has the form (m_1, m_2) , as desired.

(\Leftarrow) Now suppose that π is a full permutation of $[n]$ such that its top-level bracketing has the form (m_1, m_2) , where m_1 is a meld of length $k < n$ representing a permutation of $[k]$. It follows by definition that π is decomposable. This completes the proof of the Lemma. \square

Remark 4.1. *We note the following:*

- i. *A moment's reflection shows that the results of left bracketing (or other bracketing algorithm) on a permutation π of $\{k + 1, k + 2, \dots, k + n\}$ will be identical in structure to the results of the same bracketing algorithm on the reduced form π_{red} . For example, let $\pi = 68745$. Then*

$$\begin{aligned} \text{left bracketing of } \pi &= [(6[87])(45)], \text{ and} \\ \text{left bracketing of } \pi_{\text{red}} &= [(3[54])(12)]. \end{aligned}$$

In particular, Lemma 4.2 holds for a full permutation of $\pi = \{k + 1, k + 2, \dots, k + n\}$.

- ii. *The proof of Lemma 4.2 did not reference the tile merging or corresponding bracketing algorithm used. It follows that for full permutations π , the form (i.e., (m_1, m_2) or $[m_1, m_2]$) of π 's top-level bracketing is independent of the bracketing algorithm, although the melds m_1 and m_2 are not, as shown, for instance, by Example 2.6.*

4.2 The Half-lemma

Theorem 4.1. *For $n \geq 2$, the involution R , restricted to the set of full permutations of $[n]$, carries indecomposable permutations to decomposable permutations and vice versa.*

Proof. Let π be a full indecomposable permutation of $[n]$, for $n \geq 2$. We must show that π^R is decomposable. By Lemma 4.2, the top-level bracketing of π has the form $[m_1, m_2]$ for any bracketing algorithm (although the melds m_1 and m_2 may differ with different algorithms). Whence, by Remark 2.4, the square unitary tiles T_1 and T_2 corresponding to m_1 and m_2 that appear during right merging have the form $\begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix}$ (prior to being merged). Then, as observed in the proof of Lemma 4.1, left merging on π^R is the mirror image of right merging on π . Thus, the top-level bracketing of π^R will correspond to the mirror image of $\begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix}$, which is $\begin{pmatrix} 0 & T_1 \\ T_2 & 0 \end{pmatrix}$. Accordingly, by Lemma 4.2, π^R is decomposable. Similarly, if π is decomposable, then π^R is indecomposable. This completes the proof. \square

As an immediate consequence, we obtain

Corollary 4.1 (the Half-lemma). *For $n \geq 2$, half of the full permutations of $[n]$ are indecomposable and half are decomposable.* \square

4.3 Indecomposable components of a full permutation

Our goal in this section is to show that the indecomposable components of a full permutation π of $[n]$ can be obtained from the left bracketing of π . To this end, we prove the following Lemma:

Lemma 4.3. *Let $\pi = a_1 a_2 \dots a_n$ be a permutation of $[n]$, and let $\pi = \pi_1 \pi_2 \dots \pi_q$ be the factorization of π into indecomposable components. Then*

$$\pi_q = \text{the maximum-length indecomposable suffix of } \pi.$$

Proof. From the discussion in Section 3.2, one sees that π_1 is an indecomposable permutation of $[k]$ for some $k \leq n$; if $k < n$, one has that π_2 is an indecomposable permutation of $\{k+1, \dots, k+l\}$ for some $l \leq n-k$, and so on. In this way, we obtain that the rightmost indecomposable component $\pi_q = a_v a_{v+1} \dots a_n$ satisfies the following condition (where the LHS is taken to be $-\infty$ if $v = 1$):

$$\max(\{a_1, a_2, \dots, a_{v-1}\}) < \min(\{a_v, a_{v+1}, \dots, a_n\}).$$

It is then clear that any suffix of π strictly containing $\{a_v, a_{v+1}, \dots, a_n\}$ is necessarily decomposable, so we obtain that π_q is the maximum-length indecomposable suffix of π . \square

Lemma 4.3 leads to a way to compute the indecomposable components of a full permutation π of $[n > 1]$ through left bracketing.

Theorem 4.2. *Let $\pi = a_1 a_2 \dots a_n$ be a full permutation of $[n]$. If the top-level left bracketing of π has the form $[m_1, m_2]$, then $\pi = \pi_1$ is indecomposable. Otherwise, π is decomposable, the top-level left bracketing has the form (m_1, m_2) , and m_2 is the rightmost indecomposable component of π (with its left bracketing).*

Proof. By Lemma 4.2, we know that the top-level bracketing of a full permutation of $[n > 1]$ has the form $[m_1, m_2]$ (resp. (m_1, m_2)) if and only if π is indecomposable (resp. decomposable); in particular, the first statement holds. On the other hand, if π is decomposable, the left bracketing has the form (m_1, m_2) , where m_2 is either a singleton or has the form $[m_{21}, m_{22}]$, by Lemma 2.5. In either case, we have that every element a_j of m_1 is less than every element a_l of m_2 . Furthermore, m_2 is indecomposable: this is clear if m_2 is a singleton, and otherwise is a consequence of Lemma 4.2 and point i. of Remark 4.1; indeed, $m_2 = [m_{21}, m_{22}]$ is the bracketing of a full permutation of $\{n-s+1, n-s+2, \dots, n\}$ for some $s < n$. Accordingly, m_2 is the bracketing of the maximum-length indecomposable suffix of π , so Lemma 4.3 yields that m_2 is the (bracketed) rightmost indecomposable component of π , as desired. \square

Remark 4.2. *Given a full decomposable permutation π of $[n]$ for $n > 1$, Theorem 4.2 enables us to extract π 's indecomposable components $\pi_1 \pi_2 \dots \pi_q$ directly from its left bracketing, as follows: From the top-level bracketing (m_1, m_2) , we obtain $m_2 = \pi_q$, so $\pi = m_1 \pi_q$. If m_1 is indecomposable, then $\pi_{q-1} = \pi_1 = m_1$; otherwise $m_1 = (m_{11}, m_{12})$ and so $\pi_{q-1} = m_{12}$, and so on.*

Example 4.2. *We illustrate Remark 4.2 with the following example: $\pi = 312645798$. We first apply left bracketing:*

$$\begin{aligned} 312645798 &\mapsto 3(12)645798 &\mapsto [3(12)]645798 \\ &\mapsto [3(12)]6(45)798 &\mapsto [3(12)][6(45)]798 \\ &\mapsto ([3(12)][6(45)])798 &\mapsto ((([3(12)][6(45)])7)98) \\ &\mapsto ((([3(12)][6(45)])7)98) &\mapsto (((([3(12)][6(45)])7)98)) \end{aligned}$$

The result shows that π is full. Reading the computation backwards, we obtain:

$$\begin{aligned} \text{Rightmost component of } ((([3(12)][6(45)])7)98) &= [98], \\ \text{Rightmost component of } ((([3(12)][6(45)])7) &= 7, \\ \text{Rightmost component of } ([3(12)][6(45)]) &= [6(45)], \\ \text{Rightmost component of } [3(12)] &= [3(12)] \text{ (it's indecomposable)}. \end{aligned}$$

Hence π factors into indecomposable components as follows: $\pi = (312)(645)(7)(98)$.

The next result completes the agenda for Section 4 of the paper.

Theorem 4.3. *Let π be a permutation of $[n]$, for $n \geq 1$. Then π is full if and only if each of its indecomposable components is full.*

Proof. Let π be a permutation and $\pi_1\pi_2\ldots\pi_q$, for $q \geq 1$, its factorization into indecomposable components. If $q = 1$ (that is, π is indecomposable), the theorem is immediate, so it suffices to show the result holds if $q > 1$ ($\Leftrightarrow \pi$ is decomposable). Let m_j denote the left-bracketing of π_j , for $1 \leq j \leq q$.

(\Rightarrow) Suppose π is a full decomposable permutation of $[n]$. Then by Theorem 4.2 and Remark 4.2, we have that the left bracketing of π has the form

$$((\ldots((m_1, m_2), m_3), \ldots, m_{q-1}), m_q).$$

Recall that the left bracketing is a record of the progress of left merging beginning with the permutation matrix M_0 of π . In particular, each of the melds m_j represents a full permutation corresponding to a square unitary tile, that is, each π_k is full, as desired.

(\Leftarrow) Suppose π is a decomposable permutation of $[n]$ with factorization into indecomposable components $\pi_1\pi_2\ldots\pi_q$, for $q > 1$, and such that each factor π_j is a full permutation. Let m_j denote the left bracketing of π_j for $1 \leq j \leq q$. From the discussion at the start of the proof of Lemma 4.3, we know that there is an increasing sequence of integers

$$1 \leq s_1 < s_2 < \cdots < s_{q-1} < n$$

such that

$$\begin{aligned} m_1 &\text{ is a bracketed permutation of } \{1, 2, \ldots, s_1\}, \\ m_2 &\text{ is a bracketed permutation of } \{s_1 + 1, s_1 + 2, \ldots, s_2\}, \\ &\vdots \\ m_{q-1} &\text{ is a bracketed permutation of } \{s_{q-2} + 1, s_{q-2} + 2, \ldots, s_{q-1}\}, \\ m_q &\text{ is a bracketed permutation of } \{s_{q-1} + 1, s_{q-1} + 2, \ldots, n\}. \end{aligned}$$

One can now carry out bootstrap percolation on the permutation matrix associated to π by first percolating the tiles associated to the π_j ; since the latter are by hypothesis full permutations, one obtains a matrix containing a sequence of square unitary tiles that are diagonally adjacent along the antidiagonal, which when merged yield a single square unitary tile of size n , demonstrating that π is full. This completes the proof. \square

Example 4.3. To illustrate the second part of the preceding proof, consider the permutation $\pi = (312)(645)(7)(98)$ of Example 4.2. One sees by inspection that the (parenthesized) indecomposable components are full. In the following figure, the permutation matrix of π is shown to the left and the matrix following the percolation of the indecomposable components to the right.

$$\left(\begin{array}{cccccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \quad \left(\begin{array}{cccccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

5. Counting full permutations

Let I_n be the number of indecomposable permutations of $[n] = \{1, 2, \ldots, n\}$, let p_n be the number of full permutations of $[n]$, and let q_n be the number of full indecomposable permutations of $[n]$. The goal of this section is to give a new proof of the following theorem, the main result of [5]:

Theorem 5.1. *The full permutation matrices are enumerated by the sequence of (shifted) large Schröder numbers. That is, $p_n = S_{n-1}$.*

The key ingredient for our proof is the following consequence of Corollary 4.1:

$$\text{We have } p_1 = q_1 = 1, \text{ and for } n > 1, \text{ we have } q_n = \frac{p_n}{2}. \quad (6)$$

Proof. We begin by deriving a recursion for p_{n+2} . Let σ be a full permutation of $[n+2]$. If σ is indecomposable, we know there are q_{n+2} choices for σ . Otherwise, if σ is decomposable, write $\sigma = \mu\tau$, where μ is the first indecomposable factor and τ is the rest of the permutation. We know σ is full if and only if both μ and τ are

full; this is an easy consequence of Theorem 4.3. If μ is a permutation of $[k]$, there are q_k choices for μ and p_{n+2-k} choices for τ . Furthermore, when σ is decomposable, k must range between 1 and $n+1$, inclusive. Summing all the choices, we have the following recursion:

$$p_{n+2} = q_{n+2} + \sum_{k=1}^{n+1} q_k \cdot p_{n+2-k}.$$

Now applying (6), and breaking the term $k=1$ off from the sum, this becomes:

$$p_{n+2} = \frac{p_{n+2}}{2} + p_1 \cdot p_{n+1} + \sum_{k=2}^{n+1} \frac{p_k}{2} \cdot p_{n+2-k}.$$

Multiply through by 2 and use $p_1 = 1$ to obtain:

$$2p_{n+2} = p_{n+2} + 2p_{n+1} + \sum_{k=2}^{n+1} p_k \cdot p_{n+2-k}.$$

Thus,

$$p_{n+2} = 2p_{n+1} + \sum_{k=2}^{n+1} p_k \cdot p_{n+2-k}.$$

We can absorb one of the two terms equal to p_{n+1} back into the sum as the $k=1$ term:

$$p_{n+2} = p_{n+1} + \sum_{k=1}^{n+1} p_k \cdot p_{n+2-k}.$$

Now re-index the sum with $j = k-1$, which ranges from 0 to n :

$$p_{n+2} = p_{n+1} + \sum_{j=0}^n p_{j+1} \cdot p_{n+1-j}.$$

Let's see what this recursion says for the shifted sequence $c_{n-1} = p_n$. Rewriting in terms of c_n , our recursion now reads:

$$c_{n+1} = c_n + \sum_{j=0}^n c_j \cdot c_{n-j}.$$

This is exactly the recursion defining the (large) Schröder numbers $\{S_n\}$ [9, p. 259]. Furthermore, $1 = S_0 = c_0 = p_1$. This completes the proof of the theorem. \square

The little Schröder numbers (also called super-Catalan numbers or small Schröder numbers): 1, 1, 3, 11, 45, 197, 903, 4279, 20793, ... count, among other objects, the number of subdiagonal lattice paths from $(0,0)$ to (n,n) consisting of steps East $(1,0)$, North $(0,1)$ and Northeast $(1,1)$ with no Northeast steps along the main diagonal. See [6, A001003] for more information about this sequence.

Since the little Schröder numbers are exactly half the value of the large Schröder numbers (the initial term excepted), applying statement (6) again, we obtain:

Corollary 5.1. *The full indecomposable matrices are enumerated by the sequence of (shifted) little Schröder numbers. That is, $q_n = s_{n-1}$.*

6. The n -Kings Problem

The n -kings problem asks for the number of ways one can place n kings on an $n \times n$ chessboard, with one king in each row and column, so that none of the kings may attack each other. If you think of the chessboard as an $n \times n$ matrix, with a 1 entry where the kings are placed and other entries 0, then because of the condition of one king in each row and column, one obtains a permutation matrix. Clearly, none of the kings can attack any others if none of them are diagonally adjacent (we already know they are not horizontally or vertically adjacent since they are in distinct rows and columns), which means there are no mutable cells if bootstrap percolation is applied. Thus, the permutation matrix is in a no-growth configuration if and only if the configuration is a solution to the n -kings problem. Thus, as we stated in the Introduction, if a_n is the number of solutions

to the n -kings problem, then a_n is also the number of $n \times n$ permutation matrices that are in a ‘no-growth configuration’ with respect to bootstrap percolation. The goal of this section is to determine a_n explicitly.

First, following [5], we derive the generating function $A(t)$ of the sequence $\{a_n\}$ as the composition (3) of two known functions. Our argument depends on the main result in [7], which shows how to express the Taylor coefficients $T_n(f \circ g; a)$ of a composite function $f \circ g$ (about $t = a$) in terms of the Taylor coefficients of the ‘factors.’ (All our series involve powers of t , so $a = 0$.) Second, we will apply this same result of [7] to obtain an explicit formula for a_n from this generating function, which is a new result.

One of the known functions involved in our argument is the generating function $\varepsilon(t) = \sum_{n \geq 0} n!t^n$ of the sequence $\{n!\}$, a power series with radius of convergence 0. So, arguments that treat $\varepsilon(t)$ as a function might be invalid — this generating function only makes sense as a formal power series. The arguments in [7] relied on the function-theoretic viewpoint. However, we now have a proof of the main result of [7] based entirely on formal power series (see [8]), so the calculations we are about to do are valid.

To prove (3), we’ll count the $n \times n$ permutation matrices in two ways. On the one hand, we know there are $n!$ such matrices. On the other hand, we can count them by their final configurations after percolation. That is, we’ll compute the number of permutations that percolate to each possible final configuration, and add these counts over all such configurations.

We first consider how to count the number of permutations that percolate to a given final configuration. Recall from §2.3 that the final configuration M_r of any permutation matrix following percolation consists of m square unitary tiles $\{T_1, T_2, \dots, T_m\}$, for $1 \leq m \leq n$, in a no-growth configuration, and where the size of T_j is s_j . Fix for the moment the no-growth configuration occupied by the $\{T_j\}$. Then the possible final positions having m square unitary tiles occupying the given no-growth configuration are in one-to-one correspondence with the elements $\pi = (s_1, s_2, \dots, s_m) \in \mathcal{C}_{n,m}$ (the set of compositions of n with m parts) that specify the sizes of the tiles from left to right.

We now determine the number of $n \times n$ permutation matrices that will percolate to the final configuration corresponding to the composition π . Since the percolation takes place independently within each unitary tile (see [8]), it follows that the number of $n \times n$ permutation matrices which percolate to this particular M_r is the product of the number that percolate to each unitary tile T_j . Indeed, we know by Lemma 2.3 that each T_j is a “good” tile, meaning that it results from the percolation of a full permutation of $[s_j]$, the number of which we denote by p_{s_j} as in the previous section. It follows that the number of initial matrices that percolate to the final configuration corresponding to π is the product $\prod_{j=1}^m p_{s_j}$. Denoting the number of parts s_j of π that are equal to i by π_i , we can rewrite the previous product as $\prod_{i=1}^n (p_i)^{\pi_i}$.

Summing the previous expression over all the $\pi \in \mathcal{C}_{n,m}$, we find that the number of permutations that percolate to any final result having the current no-growth configuration is given by $\sum_{\pi \in \mathcal{C}_{n,m}} \prod_{i=1}^n (p_i)^{\pi_i}$. Hence, multiplying by the number of no-growth configurations with m tiles, we obtain that the total number of permutations that percolate to any m -tiled result is given by

$$a_m \cdot \sum_{\pi \in \mathcal{C}_{n,m}} \prod_{i=1}^n (p_i)^{\pi_i}.$$

Finally, summing over m , we find that the number of permutations of $[n]$ is given by

$$n! = \sum_{m=1}^n a_m \sum_{\pi \in \mathcal{C}_{n,m}} \prod_{i=1}^n (p_i)^{\pi_i}. \quad (7)$$

Recall $\varepsilon(t) = \sum_{n \geq 0} n!t^n$ is the ordinary generating function of the sequence $\{n!\}$, and $A(t)$ is the generating function for $\{a_n\}$. If we let $B(t)$ stand for the ordinary generating function of $\{p_n\}$, then applying the main result of [7] (which expresses the n^{th} Taylor coefficient $T_n(A \circ B; 0)$ of the composition $A \circ B$ on the LHS in terms of the Taylor coefficients $T_m(A; 0) = a_m$ of A and $T_i(B; 0) = p_i$ of B on the RHS), we obtain:

$$T_n(A \circ B; 0) = \sum_{m=1}^n a_m \sum_{\pi \in \mathcal{C}_{n,m}} \prod_{i=1}^n (p_i)^{\pi_i}.$$

Equation (7) then yields that the n^{th} Taylor coefficient of $A \circ B$ is $n!$, proving that $A(B(t)) = \varepsilon(t)$.

This argument is essentially an expansion of the heuristic argument outlined in Section 3 of [5] to obtain the composition, but we think the result in [7] makes the argument clear and precise.

Next, what is the sequence $\{p_n\}$? We saw in the last section that the number of full $n \times n$ permutation matrices is $p_n = S_{n-1}$, the $(n-1)$ 'st (large) Schröder number. Thus, $B(t) = tR(t)$, where $R(t)$ is the ordinary generating function of the Schröder numbers (denoted $R(x)$ in [5]). Combined with the above, we have:

$$A(tR(t)) = \varepsilon(t). \quad (8)$$

We know that $B(t) = tR(t) = \frac{1-t-\sqrt{1-6t+t^2}}{2}$. Define $g(t) = \frac{t(1-t)}{1+t}$. We leave it to the reader to verify that $g(t)$ and $B(t)$ are inverse functions. It follows by composing both sides of (8) by g on the right that:

$$A \circ B = \varepsilon, \text{ hence } A \circ B \circ g = \varepsilon \circ g.$$

Since $B \circ g$ is the identity function, we conclude $A = \varepsilon \circ g$:

$$A(t) = \varepsilon(g(t)) = \varepsilon\left(\frac{t(1-t)}{1+t}\right). \quad (9)$$

This is the derivation of $A(t)$ in [5], giving the generating function for the solution to the n -kings problem.

We now aim to extract our explicit solution from this. Observe that $g(0) = 0$, and by definition we have that

$$T_m(\varepsilon; 0) = m!.$$

As for the inner function, by the linearity of the operator $T_i(-; 0)$, we obtain, for $i \geq 2$:

$$T_i(g; 0) = T_i\left(\frac{t(1-t)}{1+t}; 0\right) = T_i\left(\frac{t}{1+t}; 0\right) - T_i\left(\frac{t^2}{1+t}; 0\right).$$

Now, multiplying a function by powers of t just shifts the coefficients of its series, so we can rewrite this as a linear combination of coefficients of an easily recognized geometric series:

$$\begin{aligned} T_1(g; 0) &= T_{i-1}\left(\frac{1}{1+t}; 0\right) - T_{i-2}\left(\frac{1}{1+t}; 0\right) \\ &= (-1)^{i-1} - (-1)^{i-2} \\ &= (-1)^{i-2}(-1-1) \\ &= (-1)^{i-2}(-2) = 2(-1)^{i-1}. \end{aligned}$$

For $i = 1$, we have $T_1(g; 0) = T_1\left(\frac{t}{1+t}; 0\right) = 1$, since $T_1\left(\frac{t^2}{1+t}; 0\right) = 0$. And for $i = 0$ we have already observed that $g(0) = 0$. Summarizing, we have:

$$T_i(g; 0) = \begin{cases} 0, & \text{if } i = 0, \\ 1, & \text{if } i = 1, \text{ and} \\ 2(-1)^{i-1}, & \text{if } i \geq 2. \end{cases}$$

Applying the main result of [7], we obtain:

$$\begin{aligned} a_n &= T_n(\varepsilon \circ g; 0) \\ &= \sum_{m=1}^n T_m(\varepsilon; 0) \sum_{\pi \in \mathcal{C}_{n,m}} \prod_{i=1}^n (T_i(g; 0))^{\pi_i} \\ &= \sum_{m=1}^n m! \sum_{\pi \in \mathcal{C}_{n,m}} \prod_{i=2}^n (2(-1)^{i-1})^{\pi_i}. \end{aligned}$$

Note that for any composition π , we have $T_1(g; 0)^{\pi_1} = (1)^{\pi_1} = 1$, explaining why we can ignore the term $i = 1$ in the product. Also, observe that

$$\begin{aligned} \prod_{i=2}^n (2(-1)^{i-1})^{\pi_i} &= \prod_{i=2}^n 2^{\pi_i} \prod_{i=2}^n (-1)^{(i-1)\pi_i} \\ &= 2^{\sum_{i \geq 2} \pi_i} \cdot (-1)^{\sum_{i \geq 2} i\pi_i - \sum_{i \geq 2} \pi_i} \\ &= 2^{\ell(\pi) - \pi_1} \cdot (-1)^{n - \pi_1} \cdot (-1)^{-\ell(\pi) + \pi_1} \\ &= 2^{\ell(\pi) - \pi_1} \cdot (-1)^{n - \ell(\pi)} \\ &= (-1)^n \frac{(-2)^{\ell(\pi)}}{2^{\pi_1}}. \end{aligned}$$

It follows that

$$a_n = \sum_{m=1}^n m! \sum_{\pi \in \mathcal{C}_{n,m}} (-1)^n \frac{(-2)^m}{2^{\pi_1}}.$$

Pulling the constants out of the sums, we obtain our explicit formula (first observed in 2016):

Theorem 6.1. *Let a_n be the number of solutions to the n -kings problem (that is, the number of ‘no growth’ $n \times n$ permutation matrices). Then we have the following explicit formula:*

$$\begin{aligned} a_n &= (-1)^n \sum_{m=1}^n m! (-2)^m \sum_{\pi \in \mathcal{C}_{n,m}} \frac{1}{2^{\pi_1}} \\ &= (-1)^n \sum_{\pi \in \mathcal{C}_n} \frac{\ell(\pi)! (-2)^{\ell(\pi)}}{2^{\pi_1}}. \end{aligned}$$

We illustrate the theorem for $n = 5$, using the first expression. We first list the 16 compositions of 5, grouped by length:

$$\begin{aligned} &(5), \\ &(1, 4), (2, 3), (3, 2), (4, 1), \\ &(1, 1, 3), (1, 3, 1), (3, 1, 1), (1, 2, 2), (2, 1, 2), (2, 2, 1), \\ &(1, 1, 1, 2), (1, 1, 2, 1), (1, 2, 1, 1), (2, 1, 1, 1), \\ &(1, 1, 1, 1, 1). \end{aligned}$$

Using this, we can easily compute the inner sum. When $m = 1$, we’re on the first row, with only one composition (5) for which $\pi_1 = 0$, so the inner sum in this case is 1. When $m = 2$ (second row), we obtain $2 \frac{1}{2^1} + 2 \frac{1}{2^0} = 3$. In the same way for $m = 3$ (third row), we obtain $\frac{9}{4}$ for the inner sum, and for $m = 4$ we obtain $\frac{1}{2}$ and finally when $m = 5$ we obtain $\frac{1}{32}$.

Therefore, the theorem gives:

$$\begin{aligned} a_5 &= (-1)^5 \left[1! (-2)^1 \cdot 1 + 2! (-2)^2 \cdot 3 + 3! (-2)^3 \cdot \frac{9}{4} + 4! (-2)^4 \cdot \frac{1}{2} + 5! (-2)^5 \cdot \frac{1}{32} \right] \\ &= 2 - 24 + 108 - 192 + 120 = 14, \end{aligned}$$

as expected.

Remark 6.1. Equation (9) yields a second explicit formula for a_n . The idea is to expand

$$\begin{aligned} A(t) = \varepsilon(g(t)) &= \varepsilon\left(\frac{t(1-t)}{1+t}\right) \\ &= \sum_{m=0}^{\infty} m! t^m (1-t)^m (1+t)^{-m} \\ &= \sum_{m=0}^{\infty} m! t^m \left(\sum_{q=0}^m (-1)^q \binom{m}{q} t^q \right) \left(\sum_{k=0}^{\infty} \binom{-m}{k} t^k \right) \end{aligned}$$

using the binomial theorem (for both non-negative and negative exponents), and then to extract the coefficient of t^n , which is a_n . (We thank Ira Gessel for suggesting this approach.) One obtains

$$\begin{aligned} a_n &= \sum_{m=0}^n m! \left(\sum_{q=0}^{n-m} (-1)^q \binom{m}{q} \binom{-m}{n-m-q} \right) \\ &= \sum_{m=0}^n m! \left(\sum_{q=0}^{n-m} (-1)^q \binom{m}{q} (-1)^{n-m-q} \binom{n-m-q+m-1}{n-m-q} \right)^\dagger \\ &= \sum_{m=0}^n m! \left(\sum_{q=0}^{n-m} (-1)^{n-m} \binom{m}{q} \binom{n-q-1}{n-m-q} \right). \end{aligned}$$

Rewriting the last expression by summing over $d = n - m$ instead of m , we obtain

$$a_n = \sum_{d=0}^n (n-d)! \left(\sum_{q=0}^d (-1)^d \binom{n-d}{q} \binom{n-q-1}{d-q} \right);$$

replacing d by k and q by i in the last expression, one obtains the formula for a_n found by Abramson and Moser [1] using a different approach; note that their outer sum runs from 0 to $n - 1$, which gives the correct answers for all $n \geq 1$.

References

- [1] M. Abramson and W. Moser, *Compositions, successions, and the n -kings problem*, Math. Mag. 39 (1966), no. 5, 269–273.
- [2] L. Comtet, *Sur les coefficients de l’inverse de la série formelle $\sum n!t^n$* , C. R. Acad. Sci. Paris 275 (1972), 569–572.

[†]Identity: $\binom{-r}{n} = (-1)^n \binom{n+r-1}{n}$

- [3] A. L. L. Gao, S. Kitaev, and P. B. Zhang, *On pattern avoiding indecomposable permutations*, arXiv:1605.05490v2 [math.CO], 2016.
- [4] M. Newman, *On theories with a combinatorial definition of “equivalence”*, Ann. of Math. 43 (1942), 223–243.
- [5] L. Shapiro and A. B. Stephens, *Bootstrap percolation, the Schröder numbers, and the n -kings problem*, SIAM J. Discrete Math. 4 (1991), no. 2, 275–280.
- [6] OEIS Foundation Inc. (2025), The On-Line Encyclopedia of Integer Sequences, Published electronically at <https://oeis.org>.
- [7] D. Vella, *Explicit formulas for Bernoulli and Euler numbers*, Integers: Electronic Journal of Combinatorial Number Theory 8 (2008), no. 1, 1–7.
- [8] D. Vella, *Composite generating functions*, AMS, in preparation.
- [9] J. West, *Generating trees and the Catalan and Schröder numbers*, Discrete Math. 146 (1995), 247–262.
- [10] Wikipedia, *Newman’s lemma*, https://en.wikipedia.org/wiki/Newman%27s_lemma, 2025, Accessed: 2025-07-21.