Statistics, Spinning Switches, Squares, Seating, and Solids

by

Peter O. Kagey

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Mathematics)

June 2022

I dedicate this dissertation to my brother Luke.

Miss you, bud.

Acknowledgements

To my advisor, Sami Assaf—thank you not just for your support and excitement about even my most eclectic mathematical interests, but also for your friendship, encouragement, and generosity since even before I was a fledgling graduate student.

To my committee member, Richard Arratia—for your great taste in problems and your surprising, and fun conversations. I'm going to miss having you work across the hall from me. I'm also going to miss your prize money.

To my committee member, David Kempe—I knew of you before I knew you, and it's been such a blessing having your thoughtfulness and interest throughout this process. Every interaction with you has been a gem.

I would like to thank all of my friends and colleagues, especially those who encouraged me to spend ever increasing time in KAP 500.

Finally, Sierra—I know I'm a broken record, but getting to spend half of a decade with you in graduate school has been the blessing of a lifetime. You make the world a richer place, and I couldn't be more thankful to have you in my corner. I can't wait to see what the rest of our lives have in store together!

Table of Contents

List of Tables

List of Figures

Abstract

Your dissertation abstract goes here.

Chapter 1

Introduction

This chapter talks about motivation and applications of my work, and reviews previous research in the field.

1.1 Motivations

TODO

Chapter 2

Spinning Switches

What this paper does:

1. Generalizes switches to arbitrary groups.
2. Proves a result when switches look like p-groups.
3. Finds the "correct" model: the wreath products.
4. Provides reductions for when strategies don't exist, which are easy to prove with the wreath product model.
5. Comes up with an example where something isn't a prime power. ($S_3 \wr C_2$ is nontrivial. Of course $G \wr \mathbf{1}$ also works.)

2.1 TODO

1. Provide solution to Winkler puzzle.
2. Fill out Example 2.5.1.
3. Incorporate Section 2.4 into earlier sections.
4. Provide example for first reduction. (Theorem 2.5.2)

5. Give other part of example for second reduction. (Theorem 2.5.4)

6. Example for third reduction. (Theorem 2.5.5)

7. Define/discuss minimal switching strategies?

8. Mention conjecture that most groups are 2-groups

2.2 Overview and Preliminaries

This section provides a brief history of the problem and provides the idea for the more general context. Section 2.3 models these generalizations in the context of the wreath product. Section 2.4 is where all of the references are. [TODO: put this section elsewhere] Section 2.5 allows us to prove when Player B does not have a winning strategy. Section 2.6 allows us to make a statement about games that have a prime number of possible moves. Section 2.7 gives us an example of new kinds of puzzles that have solutions. Section 2.8 gives us an example of new kinds of puzzles that have solutions.

2.2.1 History

A closely related puzzle was popularized by Martin Gardner in the February 1979 edition of his column "Mathematical Games." [**Gardner1979Problem**] He wrote that he learned of the puzzle from Robert Tappay of Toronto who "believes it comes from the U.S.S.R."

The version under consideration in this paper is first hinted at in 1993 [**Yehuda1993**]. Ehrenborg and Skinner consider something very similar, which they call the "Blind Bartender with Boxing Gloves" [**Ehrenborg1995**].

This was re-popularized in 2019 when it appeared in "The Riddler" from FiveThirtyEight [**FiveThirtyEight**].

My preferred version appears in Peter Winkler's 2004 book *Mathematical Puzzles A Connoisseur's Collection*

> Four identical, unlabeled switches are wired in series to a light bulb. The switches are simple buttons whose state cannot be directly observed, but can be changed by pushing; they are mounted on the corners of a rotatable square. At any point, you may push, simultaneously, any subset of the buttons, but then an adversary spins the square. Show that there is a deterministic algorithm that will enable you to turn on the bulb in at most some fixed number of steps. [**Winkler2004**]

TODO: Sidana paper has nice history.

2.2.2 Generalizing Switches

"The problem can also be generalized by replacing glasses with objects that have more than two positions. Hence the rotating table leads into deep combinatorial questions that as far as I know have not yet been explored." [**Gardner1979Solution**]

Switches that instead behave like n-state roulettes with a single on position are considered by Yehuda, Etzionn, and Moran in 1993 [**Yehuda1993**]. Yuri Rabinovich [**Rabinovich2022**] goes further by considering collections of switches that behave like vector spaces over finite fields. I go further still by considering switches that behave like arbitrary finite groups—or more generally still, finite quasigroups with identity.

[A schematic for a switch that looks like D_4.]

2.2.3 Generalizing Spinning

We can also consider different ways of rearranging the switches. In a 1995 paper [**Ehrenborg1995**], Ehrenborg and Skinner provide a criterion for which permutations of ordinary, 2-way switches yield a winning strategy. Rabinovich [**Rabinovich2022**] settles the problem for switches in a finite vector space.

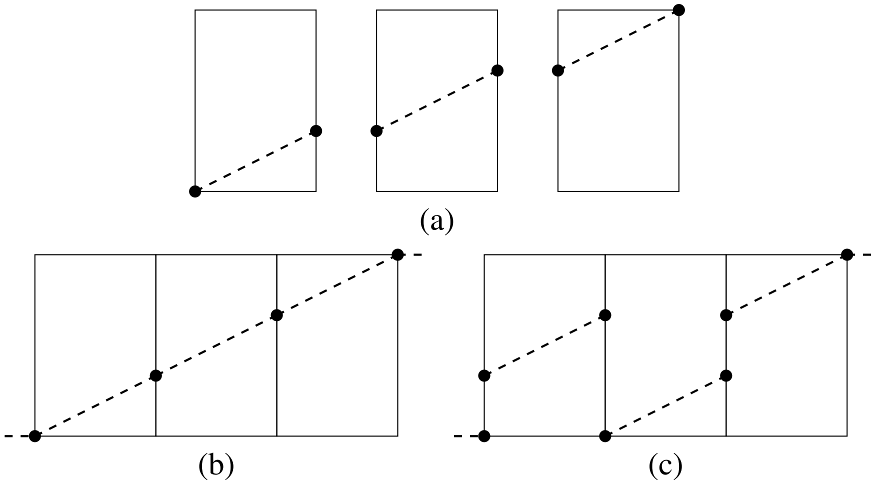


Figure 2.1: Part (a) shows a simple schematic for a switch that behaves like S_3, the symmetric group on three letters. The three rectangles can be permuted arbitrarily, but only configuration (b) completes the circuit. All other configurations fail to complete the circuit (e.g. (c)).

For example, one could imagine a "switch" that behaves like the symmetric group S_3, consisting of three identical-looking parts that need to be arranged in a particular order in order for the switch to be on.

Or one could imagine a switch that behaves like the dihedral group of the square, D_8 where the square has a single, unique orientation that completes the circuit. Or abstractly, one could think of each switch as an abstract group element, where Player B can multiply by anything they like.

2.3 The Wreath Product Model

Remind you of the definition of a wreath product, and give examples of how it models the spinning switches puzzle.

Definition 2.3.1 (Literally copied from Rotman)**.** *Let D and Q be groups, let Ω be a finite Q-set, and let $K = \prod_{\omega \in \Omega} D_\omega$, where $D_\omega \cong D$ for all $\omega \in \Omega$. Then the* ***wreath product*** *of D by Q denoted by $D \wr Q$, is the semidirect product of K by Q, where Q acts on K by $q \cdot (d_\omega) = d_{q\omega}$ for $q \in Q$ and $(d_\omega) \in \prod_{\omega \in \Omega} D_\omega$. The normal subgroup K of $D \wr Q$ is called the* ***base*** *of the wreath product.*

TODO: I prefer the Wikipedia version where $q \cdot (d_\omega) = d_{q^{-1}\omega}$

The reason this definition is used is because it models the game well, where G models the behavior of the switches, Ω models the switches themselves, and the way H acts on Ω models the ways the adversary can "spin" the board.

An element of $(k,h) \in G \wr H$ represents a turn of the game: Player B chooses k to indicate how they want to modify each of their switches and then Player A chooses k to indicate how they want to permute the switches.

Example 2.3.2. *Consider the setup in the original version of the problem consisting of two-way switches ($\mathbb{Z}_2$) on the corners of a rotating square ($C_4 \cong \langle 0^\circ, 90^\circ, 180^\circ, 270^\circ \rangle$). This can be modeled as a game on the wreath product $\mathbb{Z}_2 \wr C_4$. We will use the convention that the base of the wreath product, K is ordered upper-left, upper-right, lower-right, lower-left, and the group action is specified by degrees in the clockwise direction.*

Consider the following two turns:

1. *Turn 1:* $((1,0,1,0), 90^\circ) \in \mathbb{Z}_2 \wr C_4$.
   (a) *Player B toggles the upper-left and lower-right switches.*
   (b) *Player A rotates the table* 90° *clockwise.*
2. *Turn 2:* $((1,0,0,0), 180^\circ) \in \mathbb{Z}_2 \wr C_4$.
   (a) *Player B toggles the upper-left switch.*
   (b) *Player A rotates the table* 90° *clockwise.*

As illustrated in Figure 2.2, the net result of these two turns is the same as a single turn where Player B toggles the upper-left, upper-right, and lower-left switches and Player A rotates the board 270° *clockwise.*

The multiplication under the wreath product agrees with this:

$$((1,0,1,0), 90^\circ) \cdot ((1,0,0,0), 180^\circ) = ((1,0,1,1), 270^\circ)$$

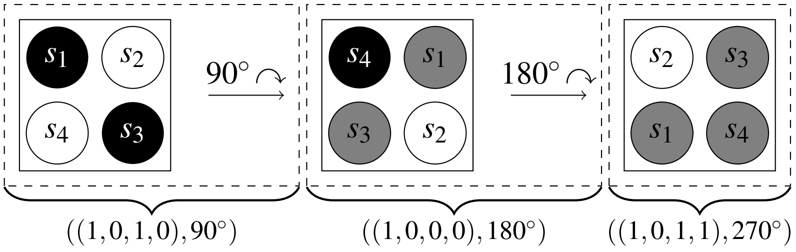


Figure 2.2: An illustration of two turns each in the Spinning Switches puzzle, modeled as elements of a wreath product.

Occasionally it is useful to designate a particular state of the switches as the on state or the winning state, and ordinarily the identity state is the choice given for this. However, the existence of a winning strategy does not depend on a particular choice in the winning state; instead, a winning strategy is equivalent to a choice of moves that will walk over all of the possible configuration states, regardless of the choice of the adversaries spin.

Definition 2.3.3. *A* ***switching strategy*** *is a finite sequence, $\{k_i \in K\}_{i=1}^{N}$, such that for every sequence $\{h_i \in H\}_{i=1}^{N}$,*

$$p(\{e_{G\wr H}, (k_1,h_1), (k_1,h_1)\cdot(k_2,h_2), \cdots, (k_1,h_1)\cdot(k_2,h_2)\cdots(k_N,h_N)\}) = K.$$

where $p\colon G\wr_{\Omega} H \to K$ is the projection map from the wreath product $G\wr_{\Omega} H$ onto its base K.

This definition is useful because it puts the problem into purely algebraic terms. It is also useful because it abstracts away the initial state of the switches: regardless of the initial state $k \in K$, a existence of a switching strategy means that its inverse $k^{-1} \in K$ appears in the sequence.

It's also worth noting that this model can be thought of as a random model or an adversarial model: the sequence $\{h_i \in H\}$ can be chosen after the sequence $\{k_i \in K\}$ in a deterministic way randomly.

1. Random model to adversarial model

2. Retroactively changes initial conditions

3. Spin

4. Proves lower bound of number of moves

2.4 Historical Progress

2.4.1 Yehuda (1993) [Roulette wheel]

Theorem 2.4.1. *The game on $\mathbb{Z}_n \wr C_m$ has a switching strategy if and only if $n = p^\alpha$ and $m = p^\beta$ where p is prime and α and β are nonnegative integers. They also deal with words with q^β letters over $\mathbb{F}_q$.*

2.4.2 Ehrenborg/Skinner (1995) [Scrambling]

Theorem 2.4.2. *Interested in fixing $G = \mathbb{Z}_2$ and looking at permutation representations of H, and determining when switching strategies exist. They look at a particular "sub-poset" of the set partitions of H partially ordered by refinement, and give a condition which is equivalent to a switching strategy.*

2.4.3 Sharma/Sidana (2021) [Other related games]

2.4.4 Yuri Rabinovich (2022)

[Switches are V over $\mathbb{F}_q$, arbitrary scrambling]

Theorem 2.4.3. *Let V be a vector space over a finite field $\mathbb{F}_q$ of characteristic p, and let V^+ be the group under addition. Let G be a group that acts linearly and faithfully on V. Then $G \wr V^+$ has a switching strategy if and only if G is a p-group.*

2.5 Reductions

There are essentially three ways to show that $G \wr H$ does not have a solution: directly, or via one of two *reductions* (or a combination thereof).

2.5.1 Puzzles Without Switching Strategies

Using results from Rabinovich [**Rabinovich2022**], we can give examples of puzzles that don't have solutions. This section allows us to take those examples and stretch them into wider families of examples.

Example 2.5.1. *The game $\mathbb{Z}_2 \wr C_3$ does not have a switching strategy. Here's how to see it...*

2.5.2 Reductions on Switches

Theorem 2.5.2. *If $G \wr H$ does not have a switching strategy and G' is a group with a quotient $G'/N \cong G$, then $G' \wr H$ does not have a switching strategy.*

Proof. I will prove the contrapositive, and suppose that $G' \wr H$ has a switching strategy $\{k_i' \in K'\}_{i=1}^N$. The quotient map $\varphi \colon G' \mapsto G$ extends coordinatewise to $\hat{\varphi} \colon K' \mapsto K$.

The sequence $\{\hat{\varphi}(k_i') \in K\}_{i=1}^N$ is a switching strategy on $G \wr H$. [Say something about how the projection map is ?linear? wrt $\hat{\varphi}$? Say ϕ induces a homomorphism from $G' \wr H$?]

Want to prove

$$p((\hat{\varphi}(k_1'), h_1) \dots (\hat{\varphi}(k_i'), h_i)) = \hat{\varphi}(p'((k_1', h_1) \dots (k_i', h_i)))$$

where $p \colon G \wr H \to K$ and $p' \colon G' \wr H \to K'$ □

Example 2.5.3. *We know that $\mathbb{Z}_2 \wr C_3$ doesn't have a switching strategy. This means that $\mathbb{Z}_6 \wr C_3$ does not have a switching strategy either.*

2.5.3 Reductions on Spinning

Theorem 2.5.4. *If $G \wr H$ does not have a switching strategy and H' is a group with a subgroup $A \leq H'$ such that $A \cong H$, then $G \wr H'$ does not have a switching strategy.*

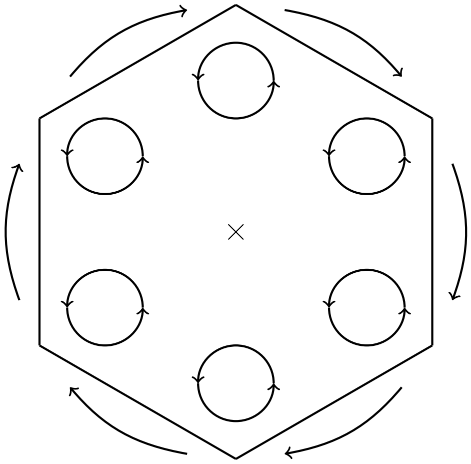


Figure 2.3: If there were a solution to $\mathbb{Z}_2 \wr C_6$, then there would be a solution to ...

Theorem 2.5.5. *(Closely related to Theorem 2.5.4) If H' is a group with a subgroup $A \leq H'$ such that $A \cong H$, Ω' is an orbit of $\omega \in \Omega$ under A, and $G \wr_{\Omega'} H$ does not have a switching strategy, then $G \wr H'$ does not have a switching strategy.*

Proof. I will also prove the contrapositive. Assume that $G \wr H'$ does have a switching strategy, $\{k_i\}_{i=1}^N$. Then by definition, for any sequence $\{h'_i\}_{i=1}^N$, the projection of the sequence

$$p(\{(k_1,h'_1)\cdot(k_2,h'_2)\cdots(k_i,h'_i)\}_{i=1}^N) = K,$$

and in particular this is true when h'_i is restricted to be in the subgroup H. Thus a switching strategy for $G \wr H'$ is also a valid switching strategy for $G \wr H$. □

TODO: we have to be careful here, because the simple proof doesn't change the number of switches, it just makes the set of "rotations" smaller. In the case of the example of $\mathbb{Z}_2 \wr C_6$, our group action is no longer transitive, but instead we have two triangular orbits.

TODO: (Something is wrong about this question, but the spirit is right) Is it true that if $G \wr_\Omega H$ doesn't work then $G \wr_{\Omega'} H'$ doesn't work where Ω' is any orbit under N?

Example 2.5.6. *We know that $\mathbb{Z}_2 \wr C_3$ doesn't have a switching strategy. This means that $\mathbb{Z}_2 \wr C_6$ does not have a switching strategy either.*

2.6 Switching Strategies on p-Groups

Theorem 2.6.1. *The wreath product $G \wr H$ has a switching strategy if there exists a normal subgroup $N \trianglelefteq G$ such that both $N \wr H$ and $G/N \wr H$ have switching strategies.*

Proof. Let $S_{G/N} = \{k_i^{G/N} \in K_{G/N}\}$ denote the switching strategy for $G/N \wr H$, and let $S_N = \{k_i^N \in K_N\}$ denote the switching strategy for $N \wr H$.

First, we partition G into $|G|/|N| = m$ cosets of N:

$$G = g_1 N \sqcup g_2 N \sqcup \cdots \sqcup g_m N.$$

From the switching strategy $\{k_i^{G/N} \in K_{G/N}\}$, we can get a sequence $S_{G'} = \{k_i' \in K_G\}$ by picking the coset representatives coordinatewise.

This sequence is not itself a switching strategy, but it does "hit" all combinations of cosets. That is, for every "spinning sequence" $\{h_i \in H\}$, and sequence of cosets $(g_{i_1}H, g_{i_2}H, \ldots, g_{i_m}H)$, there exists an index n such that

$$p((k_1', h_1) \ldots (k_n', h_n)) \in g_{i_1}H \times g_{i_2}H \times \cdots \times g_{i_m}H$$

Now if we intersperse $S_G' \circledast S_N$, this forms a switching strategy because ...

$$S_G' \circledast S_N = (\underbrace{k_1^N, k_2^N, \ldots, k_{n_N}^N}_{B_0}, \underbrace{k_1', k_1^N, k_2^N, \ldots, k_{n_N}^N}_{B_1}, \ldots, \underbrace{k_{n'}', k_1^N, k_2^N, \ldots, k_{n_N}^N}_{B_{n'}})$$

The partial products that end in block B_i all have switches in the same cosets of N, and the S_N strategy then hits all elements of K_G that belong to that combination of cosets.

□

Theorem 2.6.2. *[**Rabinovich2022**] Assume that a finite group H acts linearly and faithfully on a vector space V over a finite field $\mathbb{F}_q$ of characteristic p. Then (G,V) is friendly if and only if G is a $p-group$.*

Corollary 2.6.3. *If H is a finite group that acts faithfully on Ω, then the wreath product $G \wr H$ has a switching strategy whenever $|G| = p^n$ for some n.*

Proof. If $|G| = p^n$, then either $G \cong \mathbb{Z}_p$ or G is not simple. If $G \cong \mathbb{Z}_p$, then there exists a strategy. Otherwise, G is not simple, so choose a normal subgroup N of order $|N| = p^t$ which gives a quotient G/N with order $|G/N| = p^{n-t}$. Then the result follows by induction. □

Corollary 2.6.4. *Based on the above construction, if $|G| = p^n$, then $G \wr C_{p^\ell}$ has a palindromic strategy of length $p^{np^\ell} - 1$.*

Proof. Is this true for general $H \neq C_{p^\ell}$? □

2.7 Switching Strategies on Other Wreath Products

So far, the literature has only contained examples of spinning strategies on wreath products that are themselves p-groups: $|G \wr_\Omega H| = |G|^{|\Omega|} \cdot |H|$, where H acts faithfully.

Of course, if $H = \mathbf{1}$ is the trivial group, then $G \wr \mathbf{1} \cong G$ has a switching strategy even if G is not a p-group. (In fact, it has $(|G| - 1)!$ switching strategies!)

2.7.1 $S_n \wr C_2$

Theorem 2.7.1. *$S_n \wr C_2$ has a switching strategy.*

Proof. We start with the observation that the symmetric group can be generated by transpositions: $S_n = \langle t_1, t_2, \dots, t_N \rangle$. This means that there is a sequence of transpositions $t'_1, t'_2, \dots, t'_M$ such that $\{\mathrm{id}, t'_1, t'_1 t'_2, \dots, t'_1 t'_2 \dots t'_M\} = S_n$.KB □

Example 2.7.2. *If $a \in S_3$, let a_1 mean multiplying one of the two copies by a and a_2 mean multiplying both of the copies by a. Then the following is a strategy:*

$$
\begin{aligned}
&(12)_2(13)_2(12)_2(13)_2(12)_2\\
&\qquad\qquad\qquad\qquad (12)_1\\
&(12)_2(13)_2(12)_2(13)_2(12)_2\\
&\qquad\qquad\qquad\qquad (13)_1\\
&(12)_2(13)_2(12)_2(13)_2(12)_2\\
&\qquad\qquad\qquad\qquad (12)_1\\
&(12)_2(13)_2(12)_2(13)_2(12)_2\\
&\qquad\qquad\qquad\qquad (13)_1\\
&(12)_2(13)_2(12)_2(13)_2(12)_2\\
&\qquad\qquad\qquad\qquad (12)_1\\
&(12)_2(13)_2(12)_2(13)_2(12)_2
\end{aligned}
$$

In general, if you can walk through G with elements of order 2, then there is a strategy.

2.8 Open questions

2.8.1 Palindromic switching strategies

In all known examples, when there exists a switching strategy S, there exists a *palindromic* switching strategy $S' = \{k'_i \in K\}_{i=0}^{N}$ such that $k'_i = k'_{N-i}$ for all i.

Conjecture 2.8.1. *Whenever $G \wr H$ has a switching strategy, it also has a palindromic switching strategy.*

I'm interested in the answer even in the case of $G \wr 1 \equiv G$.

(MSE)

2.8.2 Quasigroup switches

In the paper we modeled switches as groups. This is because groups have desirable properties:

1. Closure. Regardless of which state a switch is in, modifying the state is the set of states.

2. Identity. We don't have to toggle a switch on a given turn.

3. Inverses. If a switch is off, we can always turn it on.

It turns out that we don't need the axiom of associativity, because the sequencing is naturally what computer scientists call "left associative". Thus, we can model switches a bit more generally as *loops* (i.e. quasigroups with identity.)

2.8.3 Expected number of turns

If we're to play the game uniformly at random, we're equally likely to win on any turn (of course, we never choose the "do nothing" move), so the expected number of moves is $|K| - 1$.

If there's a switching strategy, we're equally likely to win on any turn, so the expected number is $(N+1)/2$ with an N move strategy. In all cases, when a strategy is known, a *minimal* strategy is known, so this is reduced to $|K|/2$.

Conjecture 2.8.2. *Whenever $G \wr H$ has a switching strategy, it also has a minimal switching strategy.*

For setups that don't have a switching strategy, what is an (infinite) strategy that minimizes the expected number of turns? We can always do a bit better than the naive play by saying never do $(g, g, ..., g) \in K$ followed by $(g^{-1}, g^{-1}, ..., g^{-1}) \in K$.

> This puzzle reached me via Sasha Barg of the University of Mary- land, but seems to be known in many places. Although no fixed number of steps can guarantee turning the bulb on in the three-switch version, a smart randomized algorithm can get the bulb on in at most $5\frac{5}{7}$ steps on average, against any strategy by an adversary who sets the initial configuration and turns the platform. [**Winkler2021**]

2.8.4 Multiple moves between each turn

We could modify the puzzle so that the adversary's spinning sequence $\{h_i \in H\}$ is constained so that $h_i = e_H$ whenever $i \not\equiv 0 \bmod k$; that is, the adversary can only spin every k turns. For any finite setup $G \wr H$, there exists k such that Player B can win. (For example, take $k > |K|$ so that Player B can just do a walk of K.)

How can you compute the minimum k such that Player B has a strategy for each choice of $G \wr H$? This is an interesting statistic.

2.8.5 Different sorts of buttons

We could imagine a square board with different sorts of buttons—for instance one of the corners has an ordinary 2-way button and another has a 3-way button and so on. When do such setups have a switching strategy. (We can also put this problem into purely algebraic terms.) Of course, if one is a button and another is like S_3 then it's not clear how to keep them indistinguishable to Player B. In the case of the buttons, we can have $\mathbb{Z}$ act on either of them.

2.8.6 Counting switching strategies

Is there a good way to count the number of switching strategies? How about up to the action of H?

In the case of $S_3 \wr \mathbf{1}$, I counted the palindromic switching strategies, which can give a lower bound on the number of palindromic switching strategies of $S_3 \wr C_2$. (MSE)

2.8.7 Yehuda's "open game"

Yehuda has an "open game" version of the puzzle that goes like this: Everyone can see the state of the board. Player B says what moves (positionally) they want to make. Player A rotates the board however they see fit *then* applies Player B's move.

Conjecture 2.8.3. *Player B can always win by repeatedly choosing the inverse of the board.*

Example 2.8.4. *This strategy works for $\mathbb{Z}_2 \wr C_4$. Here's one example.*

- *The initial state of the board is one switch off and all of the others on.*
- *Player B says to turn on the off switch.*
- *Player A turns the board in order to turn off an adjacent switch.*
- *Player B says to turn on those two adjacent switches.*
- *Player A turns the board in order to toggle one of the switches, leaving one diagonal on and one off.*
- *Player B says to turn on those two diagonal swtiches.*
- *Player A rotates the board to instead turn off the two on switches so that all switches are off.*
- *Player B says to turn on all of the switches.*
- *Player A knows that rotating the board does not do anything, so they turn on all of the switches.*
- *Player B wins.*

This strategy also works for $\mathbb{Z}_3 \wr C_3$.

2.8.8 Generalizations of $S_3 \wr C_2$

In Example 2.7.2, we constructed a strategy for $S_n \wr C_2$, by exploiting the fact that S_n can be generated by elements of order 2.

Conjecture 2.8.5. *There exists a switching strategy for $S_n \wr C_4$.*

Conjecture 2.8.6. *There exists a switching strategy for $A_n \wr C_3$.*

The generalization of this conjecture, which is as likely to be false as it is to be true doesn't have evidence to support it.

Conjecture 2.8.7. *If G can be generated by elements of order p^n, and H is a p-group acting faithfully on the set of switches Ω, then $G \wr_\Omega H$ has a switching strategy.*

Chapter 3

Permutation Statistics

In this paper, we study permutations $\pi \in S_n$ with exactly m transpositions. In particular, we are interested in the expected value of $\pi(1)$ when such permutations are chosen uniformly at random. When n is even, this expected value is approximated closely by $(n+1)/2$, with an error term that is related to the number isometries of the $(n/2-m)$-dimensional hypercube that move every face. Furthermore, when $k \mid n$, this construction generalizes to allow us to compute the expected value of $\pi(1)$ for permutations with exactly m k-cycles. In this case, the expected value has an error term which is related instead to the number derangements of the generalized symmetric group $S(k, n/k-m)$.

When k does not divide n, the expected value of $\pi(1)$ is precisely $(n+1)/2$. Indirectly, this suggests the existence of a reversible algorithm to insert a letter into a permutation which preserves the number of k-cycles, which we construct.

3.1 Background

In 2010, Mark Conger [1] proved that a permutation with k descents has an expected first letter of $\pi(1) = k+1$, independent of n. This paper has the same premise, but with a different permutation statistic: the number of k-cycles of a permutation.

This section, (Section 3.1) provides an overview of where we're headed, and includes an critical example that will hopefully spark the reader's curiosity and motivate the remainder of the paper.

Section 3.2 establishes some recurrence relations for the number of permutations in S_n with a given number of k-cycles. It also contains a theorem that gives an explicit way to compute the expected value of the first letter based on these counts.

Section 3.3 describes an explicit correspondence between k-cycles of permutations in S_{kn} and fixed points of elements of the generalized symmetric group $(\mathbb{Z}/k\mathbb{Z}) \wr S_n$. Using generating functions and results from the previous section, this shows that the expected value of $\pi(1)$ of a permutation with a given number of k-cycles is intimately connected to the number of derangements of a generalized symmetric group.

While Section 3.3 emphasizes the case of S_{kn}, Section 3.4 looks at S_N where $k \nmid N$. Here, the expected value of $\pi(1)$ is simply $(N+1)/2$, which agrees with the expected value of the first letter of a uniformly chosen N-letter permutation with no additional restrictions. This fact together with the main theorem from Section 3.2 implies the existence of a bijection $\varphi_k \colon S_{N-1} \times [N] \to S_N$ that preserves the number of k-cycles whenever $k \nmid N$. Section 3.4 constructs such a bijection explicitly, and proves that it has the desired properties.

3.1.1 Motivating Examples

In support of the first examples, we start by defining the first bit of notation.

Definition 3.1.1. *Let $C_k(n,m)$ denote the number of permutations $\pi \in S_n$ such that π has exactly m k-cycles.*

These theorems—and many of the following lemmas—were discovered by looking at examples such as the following, written in both one-line and cycle notation:

Example 3.1.2. *There are $C_2(4,0) = 15$ permutations in S_4 with no 2-cycles:*

$$\begin{array}{llll}
1234 = (1)(2)(3)(4) & 2314 = (312)(4) & 3124 = (321)(4) & 4123 = (4321) \\
1342 = (1)(423) & 2341 = (4123) & 3142 = (4213) & 4132 = (421)(3) \\
1423 = (1)(432) & 2413 = (4312) & 3241 = (2)(413) & 4213 = (2)(431) \\
 & 2431 = (3)(412) & 3421 = (4132) & 4312 = (4231)
\end{array}$$

There are $C_2(4,1) = 6$ permutations in S_4 with exactly one 2-cycle:

$$\begin{array}{ll}
1243 = (1)(2)(43) & 2134 = (21)(3)(4) \\
1324 = (1)(32)(4) & 3214 = (2)(31)(4) \\
1432 = (1)(3)(42) & 4231 = (2)(3)(41)
\end{array}$$

And there are $C_2(4,2) = 3$ permutations in S_4 with exactly two 2-cycles,

$$2143 = (21)(43) \qquad 3412 = (31)(42) \qquad 4321 = (32)(41).$$

By averaging the first letter over these examples, we can compute that

$$\begin{aligned}
\mathbb{E}[\pi(1) \,|\, \pi \in S_4 \textit{ has no } 2\textit{-cycles}] &= \frac{3(1)+4(2+3+4)}{15} = \frac{13}{5}, \\
\mathbb{E}[\pi(1) \,|\, \pi \in S_4 \textit{ has exactly } 1\ 2\textit{-cycle}] &= \frac{3(1)+(2+3+4)}{6} = 2, \textit{ and} \\
\mathbb{E}[\pi(1) \,|\, \pi \in S_4 \textit{ has exactly } 2\ 2\textit{-cycles}] &= \frac{2+3+4}{3} = 3.
\end{aligned}$$

The table in Figure 3.1 gives the expected value of $\pi(1)$ given that $\pi \in S_n$ and has exactly m 2-cycles in its cycle decomposition. Notice that when i is odd, row i has a constant value of $(i+1)/2$. Also notice that the number in position (i, j) has the same denominator as the number in position $(i+2, j+1)$, and that these denominators increase with n. The sequence of denominators begins

$$1, 5, 29, 233, 2329, 27949, \ldots, \tag{3.1}$$

which agrees with the type B derangement numbers, sequence A000354 in the On-Line Encyclopedia of Integer Sequences (OEIS) [2]. In other words, the denominators in the table appear to be related to the symmetries of the hypercube that move every facet.

		m						
		0	1	2	3	4	5	6
n	1	1/1						
	2	1/1	2/1					
	3	2/1	2/1					
	4	13/5	2/1	3/1				
	5	3/1	3/1	3/1				
	6	101/29	18/5	3/1	4/1			
	7	4/1	4/1	4/1	4/1			
	8	1049/233	130/29	23/5	4/1	5/1		
	9	5/1	5/1	5/1	5/1	5/1		
	10	12809/2329	1282/233	159/29	28/5	5/1	6/1	
	11	6/1	6/1	6/1	6/1	6/1	6/1	
	12	181669/27949	15138/2329	1515/233	188/29	33/5	6/1	7/1
	13	7/1	7/1	7/1	7/1	7/1	7/1	7/1

Figure 3.1: A table of the expected value of the first letter of $\pi \in S_n$ with exactly m 2-cycles, $\mathbb{E}[\pi(1) \,|\, \pi \in S_n \text{ has exactly } m \text{ 2-cycles}]$.

3.2 Structure of permutations with m k-cycles

This section is about connecting the number of permutations with a given number of k-cycles to the expected value of the first letter. Saying this, it is appropriate to start with a 1944 theorem of Goncharov that, by the principle of inclusion/exclusion, gives an explicit formula that counts the number of such permutations.

3.2.1 Counting permutations based on cycles

Theorem 3.2.1 ([3], [4])**.** *The number of permutations in S_n with exactly m k-cycles is given by the following sum, via the principle inclusion/exclusion:*

$$C_k(n,m) = \frac{n!}{m!k^m} \sum_{i=0}^{\lfloor n/k \rfloor - m} \frac{(-1)^i}{i!\,k^i}. \tag{3.2}$$

Corollary 3.2.2. *For $k \nmid n$, there are exactly n times as many permutations in S_n with exactly m k-cycles than there are in S_{n-1}. When $k \mid n$, there is an explicit formula for the difference.*

$$C_k(n,m) - nC_k(n-1,m) = \begin{cases} 0 & k \nmid n \qquad (3.3a) \\ \dfrac{n!(-1)^{\frac{n}{k}-m}}{(n/k)!\,k^{\frac{n}{k}}} \dbinom{n/k}{m} & k \mid n \qquad (3.3b) \end{cases}$$

Proof. When $k \nmid n$, $\left\lfloor \frac{n}{k} \right\rfloor = \left\lfloor \frac{n-1}{k} \right\rfloor$, so the bounds on the sums are identical and the result follows directly

$$\frac{n!}{m!k^m} \sum_{i=0}^{\lfloor n/k \rfloor - m} \frac{(-1)^i}{i!\,k^i} - n \left(\frac{(n-1)!}{m!k^m} \sum_{i=0}^{\lfloor (n-1)/k \rfloor - m} \frac{(-1)^i}{i!\,k^i} \right) = 0. \tag{3.4}$$

Otherwise, when $k \mid n$, $\left\lfloor \frac{n-1}{k} \right\rfloor = \frac{n}{k} - 1$, so

$$\begin{aligned}
\frac{n!}{m!k^m} \sum_{i=0}^{n/k-m} \frac{(-1)^i}{i!\,k^i} - n\left(\frac{(n-1)!}{m!k^m} \sum_{i=0}^{n/k-1-m} \frac{(-1)^i}{i!\,k^i}\right) \\
&= \frac{n!}{m!k^m}\left(\frac{(-1)^{n/k-m}}{(n/k-m)!k^{n/k-m}}\right) \\
&= \frac{n!(-1)^{n/k-m}}{(n/k-m)!m!k^{n/k}} \\
&= \frac{n!(-1)^{\frac{n}{k}-m}}{(n/k)!\,k^{n/k}}\binom{n/k}{m}. \qquad (3.5)
\end{aligned}$$

□

See Section 3.4 for a bijective proof of Equation 3.3a.

3.2.2 Permutations by first letter

In order to compute the expected value of the first letter of a permutation, it is useful to be able to compute the number of permutations that have a given number of k-cycles *and* a given first letter.

Definition 3.2.3. *Let $C_k^{(a)}(n,m)$ be the number of permutations $\pi \in S_n$ that have exactly m k-cycles and $\pi(1) = a$.*

The expected value of $\pi(1)$ with a given number of k-cycles is

$$\mathbb{E}[\pi(1) \,|\, \pi \in S_n \text{ has exactly } m \text{ } k\text{-cycles}] = \frac{1}{C_k(n,m)} \sum_{a=1}^{n} aC_k^{(a)}(n,m). \qquad (3.6)$$

The following three lemmas compute $C_k^{(a)}(n,m)$ from $C_k(n,m)$.

Proposition 3.2.4. *For all $k > 1$, the number of permutations in S_n starting with* 1 *and having m k-cycles is equal to the number of permutations in S_{n-1} with m k-cycles:*

$$C_k^{(1)}(n,m) = C_k(n-1,m). \tag{3.7}$$

Proof. The straightforward bijection from $\{\pi \in S_n : \pi(1) = 1\}$ to S_{n-1} given by deleting 1 and relabeling preserves the number of k-cycles for $k > 1$. □

Proposition 3.2.5. *For all $a, b \geq 2$, the number of permutations having k-cycles and starting with a are the same as the number of those starting with b:*

$$C_k^{(2)}(n,m) = \cdots = C_k^{(a)}(n,m) = \cdots = C_k^{(b)}(n,m) = \cdots = C_k^{(n)}(n,m). \tag{3.8}$$

Proof. Since the permutations under consideration do not fix 1, conjugation by (ab) is an isomorphism which takes all words starting with a to words starting with b without changing the cycle structure. □

Lemma 3.2.6. *For all* $2 \leq a \leq n$,

$$C_k^{(a)}(n,m) = \frac{C_k(n,m) - C_k(n-1,m)}{n-1}. \tag{3.9}$$

Proof. Since

$$C_k(n,m) = C_k^{(1)}(n,m) + C_k^{(2)}(n,m) + \cdots + C_k^{(n)}(n,m), \tag{3.10}$$

using Proposition 3.2.5, for the last $(n-1)$ terms, this can be rewritten as

$$C_k(n,m) = C_k^{(1)}(n,m) + (n-1)C_k^{(a)}(n,m). \tag{3.11}$$

Solving for $C_k^{(a)}(n,m)$ and using the substitution from Proposition 3.2.4 gives the desired result.

□

Now, equipped with explicit formulas for $C_k^{(a)}(n,m)$ and $C_k(n,m)$, we can compute the expected value of $\pi(1)$ for $\pi \in S_n$ with exactly m k-cycles.

3.2.3 Expected value of first letter

Theorem 3.2.7. *For $k > 1$, the expected value of the first letter of a permutation $\pi \in S_n$ with m k-cycles is given by*

$$\begin{aligned}&\mathbb{E}[\pi(1) \mid \pi \in S_n \textit{ has exactly } m \textit{ } k\textit{-cycles}] \\ &\qquad = \frac{n}{2}\left(1 - \frac{C_k(n-1,m)}{C_k(n,m)}\right) + 1. \end{aligned} \tag{3.12}$$

Proof. Using Proposition 3.2.5, we can consolidate all but the first term of the sum in Equation 3.6

$$\sum_{a=1}^{n} aC_k^{(a)}(n,m) \tag{3.13}$$

$$= C_k^{(1)}(n,m) + \sum_{a=2}^{n} aC_k^{(n)}(n,m) \tag{3.14}$$

$$= C_k^{(1)}(n,m) + \frac{(n-1)(n+2)}{2}C_k^{(n)}(n,m) \tag{3.15}$$

$$= C_k(n-1,m) + \frac{(n-1)(n+2)}{2}\left(\frac{C_k(n,m) - C_k(n-1,m)}{n-1}\right) \tag{3.16}$$

$$= \left(\frac{n}{2}+1\right)C_k(n,m) - \frac{n}{2}C_k(n-1,m). \tag{3.17}$$

Dividing by $C_k(n,m)$ yields the result. □

Corollary 3.2.8. *When $k \nmid n$, $C_k(n,m) = nC_k(n-1,m)$ by Equation 3.3a, so*

$$\mathbb{E}[\pi(1) \mid \pi \in S_n \textit{ has exactly } m \textit{ } k\textit{-cycles}] = \frac{n}{2}\left(1 - \frac{1}{n}\right) + 1 = \frac{n+1}{2}. \tag{3.18}$$

Together with Theorem 3.2.1, this theorem and its corollary provides our first formula for the expected value of $\pi(1)$ that performs exponentially better than brute force.

3.2.4 Identities for counting permutations with given cycle conditions

Both in practical terms (if computing the expected value of $\pi(1)$ by hand or optimizing an algorithm) and in a theoretical sense, the following recurrence is simple and useful.

Lemma 3.2.9. *For $n < mk$ or $m < 0$, $C_k(n,m) = 0$. Otherwise, for all $k,m \geq 1$*

$$mC_k(n,m) = (k-1)!\binom{n}{k}C_k(n-k,m-1). \tag{3.19}$$

While this can be proven directly by the algebraic manipulation of the identity in Theorem 3.2.1, a bijective proof has been included here because it is natural and may be of interest.

Proof. Let

$$\mathscr{C}_k(n,m) = \{\pi \in S_n \mid \pi \text{ has exactly } m\ k\text{-cycles}\}. \tag{3.20}$$

Then consider the two sets, whose cardinalities match the left- and right-hand sides of the equation above:

$$X^L_{n,m,k} = \{(\pi,c) \mid \pi \in \mathscr{C}_k(n,m), c \text{ a distinguished } k\text{-cycle of } \pi\}. \tag{3.21}$$

$$X^R_{n,m,k} = \{(\sigma,d) \mid \pi \in \mathscr{C}_k(n-k,m-1), d \text{ an } n\text{-ary necklace of length } k\}. \tag{3.22}$$

The first set, $X^L_{n,m,k}$, is constructed by taking a permutation in $\mathscr{C}_k(n,m)$ and choosing one of its m k-cycles to be distinguished, so $\#X^L_{n,m,k} = mC_k(n,m)$.

In the second set, $X^R_{n,m,k}$, the two parts of the tuple are independent. There are $C_k(n-k,m-1)$ choices for the permutation σ and $(k-1)!\binom{n}{k}$ choices for the necklace d. Thus $\#X^R_{n,m,k} = (k-1)!\binom{n}{k}C_k(n-k,m-1)$.

Now, consider the map $\varphi\colon X^L_{n,m,k} \to X^R_{n,m,k}$ which removes the distinguished k-cycle and relabels the remaining $n-k$ letters as $\{1,2,\ldots,n-k\}$, preserving the relative order:

$$(\pi_1\pi_2\cdots\pi_\ell,\pi_i) \overset{\varphi}{\mapsto} (\pi'_1\pi'_2\cdots\pi'_{i-1}\pi'_{i+1}\cdots\pi'_\ell,\pi_i) \tag{3.23}$$

where π_i' is π_i after relabeling.

By construction, σ has one fewer k-cycle and k fewer letters than π.

The inverse map is similar. To recover π, increment the letters of σ appropriately and add the necklace d back in as the distinguished cycle. Thus φ is a bijection and $\#X^L_{n,m,k} = \#X^R_{n,m,k}$. □

Example 3.2.10. *Suppose* $\pi = (423)(\mathbf{61})(75)$ *in cycle notation with* (61) *distinguished. Then*

$$\varphi((423)(61)(75),(61)) = ((312)(54),(61)) \tag{3.24}$$

under the bijection φ*, described in the proof of Lemma 3.2.9.*

The recurrence in Lemma 3.2.9 suggests that understanding $C_k(n,m)$ is related to understanding $C_k(n-km,0)$, the permutations of S_{n-km} with no k-cycles. On the other hand, Corollary 3.2.2 suggests that the case where $k \mid n$ has some of the most intricate structure. We can, of course, combine these two observations and analyze the case of $C_k(kn,0)$, which has a particularly simple generating function, which will show up again in a different guise.

Lemma 3.2.11. *For* $k \geq 2$*,*

$$\sum_{n=0}^{\infty} \frac{C_k(kn,0)k^n}{(kn)!}x^n = \frac{\exp(-x)}{1-kx}. \tag{3.25}$$

Proof. By substitution of $C_k(kn,0)$ via the identity in Theorem 3.2.1,

$$\begin{aligned}
\sum_{n=0}^{\infty} \frac{C_k(kn,0)k^n}{(kn)!}x^n &= \sum_{n=0}^{\infty}\sum_{i=0}^{n} \frac{(-1)^i}{k^i i!}k^n x^n && (3.26)\\
&= \sum_{n=0}^{\infty}\sum_{i=0}^{n} \frac{(-x)^i}{i!}(kx)^{n-i} && (3.27)\\
&= \left(\sum_{n=0}^{\infty} \frac{(-x)^n}{n!}\right)\left(\sum_{n=0}^{\infty}(kx)^n\right) && (3.28)\\
&= \frac{\exp(-x)}{1-kx}. && (3.29)
\end{aligned}$$

□

This section allowed for the practical computation of the expected value of $\pi(1)$ with a given number of k-cycles, but leaves the observation about Figure 3.1 unexplained. The following section will explain the connection between the expected values of $\pi(1)$ and the facet-derangements of the hypercube.

3.3 Connection with the generalized symmetric group

This section explains the connection between the expected value of $\pi(1)$ given that π has exactly m 2-cycles and the facet-derangements of the hypercube, by telling the more general story of derangements of the generalized symmetric group. Thus it is appropriate to start this section by defining both the generalized symmetric group and its derangements.

3.3.1 Derangements of the generalized symmetric group

Definition 3.3.1. *The* ***generalized symmetric group*** $S(k,n)$ *is the wreath product* $(\mathbb{Z}/k\mathbb{Z}) \wr S_n$*, which in turn is a semidirect product* $(\mathbb{Z}/k\mathbb{Z})^n \rtimes S_n$*.*

A natural way of thinking about the symmetric group S_n is by considering how the elements act on length-n sequences by permuting the indices. Informally, we can think about the generalized symmetric group $S(k,n)$ in an essentially similar way: each element consists of an ordered pair in $(\mathbb{Z}/k\mathbb{Z})^n \rtimes S_n$, where $(\mathbb{Z}/k\mathbb{Z})^n$ gives information about what to add componentwise, and S_n gives information about how to rearrange afterward.

Example 3.3.2. *Consider the generalized permutation*

$$(\underbrace{(1,3,0)}_{\in(\mathbb{Z}/4\mathbb{Z})^3}, \underbrace{(23)}_{\in S_3}) \in S(4,3).$$

It acts on the sequence $(0,1,1) \in (\mathbb{Z}/2\mathbb{Z})^3$ *first by adding element-wise, and then permuting:*

$$\underbrace{((1,3,0),(23))}_{\in S(k,n)} \cdot (0,1,1) = \underbrace{(23)}_{\in S_3} \cdot (1+0,3+1,0+1) = (23)\cdot(1,0,1) = (1,1,0). \tag{3.30}$$

When $k=1$, the sequence $(\mathbb{Z}/1\mathbb{Z})^n$ is trivially the zero sequence, so $S(1,n) \cong S_n$. When $k=2$, $S(2,n)$ is the hyperoctahedral group that we brushed up against in Figure 3.1: the group of symmetries of the n-dimensional hypercube. When $k \geq 3$, $S(k,n)$ does not have such an immediate geometric interpretation, but it is precisely the right analog for the expected value of $\pi(1)$ when π has a given number of k-cycles.

Definition 3.3.3. *A* ***derangement*** *or* ***fixed-point-free element*** *of the generalized symmetric group is an element* $((x_1,\ldots,x_n),\pi) \in S(k,n)$ *such that for all i, either* $\pi(i) \neq i$ *or* $x_i \neq 0$.

That is, when a derangement acts on a sequence in the manner described above, it changes the position or the value of every term in the sequence. When $k=1$ and $S(1,n) \cong S_n$, this recovers the usual sense of a derangement in S_n: a permutation with no fixed points. In terms of the hyperoctahedral group, $S(2,n)$, a derangement is a symmetry of the n-cube that moves each $(n-1)$-dimensional face.

Example 3.3.4. *The element* $((1,3,0),(23)) \in S(4,3)$ *is a derangement because it increments the first term and swaps the second and third terms—thus changing the position or value for each term.*

The number of derangements of the generalized symmetric group can be described by an explicit sum via the principle of inclusion/exclusion, and it has a particularly elegant exponential generating function.

Theorem 3.3.5 ([5])**.** *For* $k>1$*, the number of derangements of the generalized symmetric group* $S(k,n)$ *is*

$$D(k,n) = k^n n! \sum_{i=0}^{n} \frac{(-1)^i}{k^i i!}. \tag{3.31}$$

which has exponential generating function

$$\sum_{n=0}^{\infty} \frac{D(k,n)}{n!} x^n = \frac{\exp(-x)}{1-kx}. \tag{3.32}$$

Notice that this agrees identically with the generating function in Lemma 3.2.11, which is our first hint in explaining the connection between k-cycles in permutations and fixed points in elements of the generalized symmetric group.

3.3.2 Permutation cycles and derangements

Lemma 3.3.6. *For $k \geq 1$, the number of permutations with $kn+km$ letters and m k-cycles is*

$$C_k(k(n+m),m) = \binom{kn+km}{kn} C_k(kn,0) \frac{(km)!}{k^m m!}. \tag{3.33}$$

Algebraic proof. This will proceed by induction on m. The base case is clear when $m = 0$, so suppose that the lemma is true up to $m-1$, that is

$$C_k(k(n+m-1),m-1) = \frac{(km-k)!}{k^{m-1}(m-1)!} \binom{kn+km-k}{kn} C_k(kn,0). \tag{3.34}$$

$$= \frac{(kn+km-k)!}{k^{m-1}(m-1)!(kn)!} C_k(kn,0). \tag{3.35}$$

Rearranging Lemma 3.2.9,

$$C_k(k(n+m),m) = \frac{(k-1)!}{m} \binom{k(n+m)}{k} C_k(k(n+m-1),m-1) \tag{3.36}$$

$$= \frac{(kn+km)!}{km(kn+km-k)!} C_k(k(n+m-1),m-1). \tag{3.37}$$

Now, notice there is a $(kn+km-k)!$ term in the numerator of Equation 3.35 and the denominator of Equation 3.37, so substituting and simplifying yields

$$C_k(k(n+m),m) = \frac{(kn+km)!}{k^m m!(kn)!} C_k(kn,0), \tag{3.38}$$

as desired. □

Combinatorial proof. This lemma lends itself to a combinatorial proof. The left hand side of the equation counts the number of permutations in S_{kn+km} with exactly m k-cycles. The right hand side of the equation says that this is the number of ways to choose kn letters in the permutation that will not be in k-cycles, and for each of these, there are $C_k(kn,0)$ ways to arrange these such that they have no k-cycles. This leaves over km letters, of which there are $(km)!/(k^m m!)$ ways to write them as products of m disjoint k-cycles. □

The following lemma uses the above identities to establish that the proportion of permutations in the symmetric group S_{kn} with exactly m k-cycles is equal to the proportion of elements in the generalized symmetric group $S(k,n)$ with exactly m fixed points.

Lemma 3.3.7. *For $k \geq 2$,*

$$\frac{C_k(kn,m)}{(kn)!} = \binom{n}{m} \frac{D(k,n-m)}{k^n n!}. \tag{3.39}$$

Proof. By solving for $D(k,n-m)$ on the right hand side and substituting $n+m$ for n, it is enough to show that the exponential generating function for $D(k,n)$ (as shown in Theorem 3.3.5) is also the exponential generating function for

$$C_k(kn+km,m) \frac{m!n!k^{n+m}}{(kn+km)!}. \tag{3.40}$$

By the identity in Lemma 3.3.6,

$$\begin{aligned}
&\sum_{n=0}^{\infty} C_k(kn+km,m)\frac{m!n!k^{n+m}}{(kn+km)!}\frac{x^n}{n!} && (3.41)\\
&\qquad = \sum_{n=0}^{\infty} \frac{(km)!}{m!k^m}\binom{kn+km}{kn} C_k(kn,0)\frac{m!n!k^{n+m}}{(kn+km)!}\frac{x^n}{n!} && (3.42)\\
&\qquad = \sum_{n=0}^{\infty} C_k(kn,0)\frac{k^n x^n}{(kn)!} && (3.43)\\
&\qquad = \frac{\exp(-x)}{1-kx}, && (3.44)
\end{aligned}$$

with the final equality being the identity in Lemma 3.2.11. □

3.3.3 Expected value of letters of permutations

We now have the ingredients we need to prove the pattern that we observed in Figure 3.1 that purported to show a relationship between permutations given number of 2-cycles and derangements of the hyperoctahedral group. These ingredients come together in the following theorem, which establishes the more general relationship between permutations with a given number of k-cycles and derangements of the generalized symmetric group, $S(k,n)$.

Theorem 3.3.8. *The expected value of the first letter of a permutation $\pi \in S_{kn}$ with exactly m k-cycles, where $k > 1$ and $0 \le m \le n$, is*

$$\mathbb{E}[\pi(1) \,|\, \pi \in S_{kn} \text{ has exactly } m \text{ } k\text{-cycles}] = \frac{kn+1}{2} + \frac{(-1)^{n-m}}{2D(k,n-m)} \tag{3.45}$$

where $D(k,n)$ is the number of derangements of the generalized symmetric group $S(k,n) = (\mathbb{Z}/m\mathbb{Z}) \wr S_n$.

Proof. Inverting the identity in Lemma 3.3.7, yields

$$\frac{\frac{(kn)!}{n!k^n}\binom{n}{m}}{C_k(kn,m)} = \frac{1}{D(k,n-m)}. \tag{3.46}$$

Multiplying through by $(-1)^{n-m}$ to match the right hand side of Equation 3.45, together with some small manipulations yields

$$1-\frac{C_k(kn,m)-(-1)^{n-m}\frac{(kn)!}{n!k^n}\binom{n}{m}}{C_k(kn,m)}=\frac{(-1)^{n-m}}{D(k,n-m)}. \tag{3.47}$$

Now adding $kn+1$ and dividing by 2 yields

$$\begin{aligned}\frac{kn}{2}\left(1-\frac{C_k(kn,m)-(-1)^{n-m}\frac{(kn)!}{n!k^n}\binom{n}{m}}{knC_k(kn,m)}\right)+1&\\ =\frac{kn+1}{2}+\frac{(-1)^{n-m}}{2D(k,n-m)},&\end{aligned} \tag{3.48}$$

which gives the right hand side as desired. Since the numerator on the left hand side is equal to $knC_k(kn-1,m)$ by Equation 3.2.2, the proof then follows from by Theorem 3.2.7. □

With the expected value of the first letter found, we can generalize this one more step to find the expected value of the i-th letter of these permutations.

Corollary 3.3.9. *The expected value of the i-th letter of a permutation in S_{kn} with exactly m k-cycles, where $n \in \mathbb{N}_{>0}$, $k > 1$, $1 \leq i \leq kn$, and $0 \leq m \leq n$, is*

$$\mathbb{E}[\pi(i)\,|\,\pi \in S_{kn} \textit{ has exactly m k-cycles}]=\frac{kn+1}{2}+\frac{(-1)^{n-m}}{2D(k,n-m)}\frac{kn+1-2i}{kn-1}.$$

Proof. Denote by N the number of permutations in S_{kn} with m k-cycles where 1 is a fixed point; denote by M the number of permutations in S_{kn} with m k-cycles where $\pi(1) = a \neq 1$. Note that while N and M implicitly depend on m, n, and k, M does not depend on a by Proposition 3.2.5.

Thus

$$
\begin{aligned}
&\mathbb{E}[\pi(1) \,|\, \pi \in S_{kn} \text{ has exactly } m \text{ } k\text{-cycles}] \\
&\qquad = \frac{1}{N+(kn-1)M}\left(N+\sum_{a=2}^{kn} aM\right) \\
&\qquad = \frac{1}{N+(kn-1)M}\left(N+\left(\frac{kn(kn+1)}{2}-1\right)M\right). \qquad (3.49)
\end{aligned}
$$

More generally, if we conjugate with $(1i)$ then

$$
\begin{aligned}
&\mathbb{E}[\pi(i) \,|\, \pi \in S_{kn} \text{ has exactly } m \text{ } k\text{-cycles}] \\
&\qquad = \frac{1}{N+(kn-1)M}\left(N+\sum_{a\neq i} aM\right) \\
&\qquad = \frac{1}{N+(kn-1)M}\left(iN+\left(\frac{kn(kn+1)}{2}-i\right)M\right). \qquad (3.50)
\end{aligned}
$$

We can extend the function $\mathbb{E}[\pi(i) \,|\, \pi \in S_{kn}$ has exactly m k-cycles$]$ to a function $f(n,k,m,i)$ where $i \in \mathbb{Q}$ is not necessarily an integer. As can be seen in Equation 3.50, f is affine function in i. By Theorem 3.3.8, when $i = 1$,

$$f(n,k,m,1) = \frac{kn+1}{2} + \frac{(-1)^{n-m}}{2D(k,n-m)}.$$

When $i = (kn+1)/2$ yields

$$f(n,k,m,(kn+1)/2) = \frac{kn+1}{2}.$$

Because $f(n,k,m,i)$ is affine in i, it is enough to use linear interpolation and extrapolation to compute f for arbitrary i. This can be done by scaling the $\frac{(-1)^{n-m}}{2D(k,n-m)}$ term by an affine function of i which is 1 when $i = 1$ and which vanishes when $i = (kn+1)/2$, namely $\frac{kn+1-2i}{kn-1}$, as desired.

□

Example 3.3.10. *For $n = 2$, $k = 2$, and $m = 0$ the expected value of the first letter in a permutation in $S_{nk} = S_4$ with no $k = 2$-cycles is $\frac{13}{5}$, as shown in Example 3.1.2. This agrees with Theorem 3.3.8:*

$$\frac{kn+1}{2} + \frac{(-1)^{n-m}}{2D(k,n-m)} = \frac{4+1}{2} + \frac{(-1)^{2-0}}{2D(2,2-0)} = \frac{5}{2} + \frac{1}{10} = \frac{13}{5}, \tag{3.51}$$

since $D(2,2) = 5$ as illustrated in Figure 3.2.

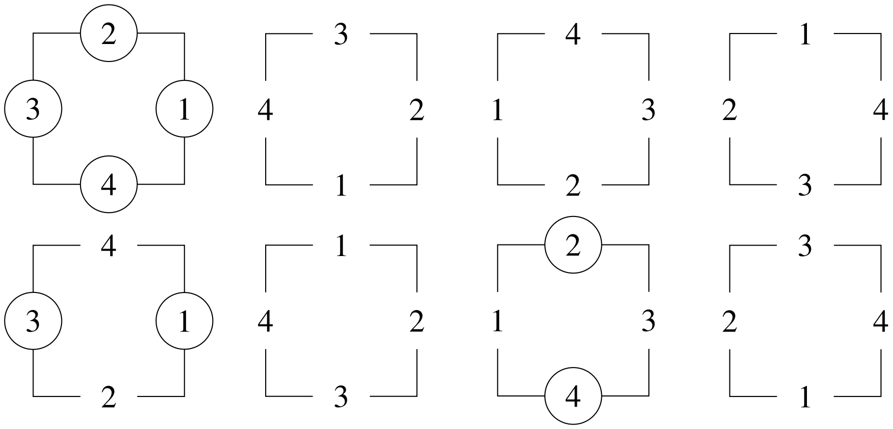


Figure 3.2: The $2^2 2! = 8$ symmetries of a square with fixed sides circled. The square (2-dimensional hypercube) has symmetry group $S(2,2) = (\mathbb{Z}/2\mathbb{Z}) \wr \mathbb{S}_2$ and $D(2,2) = 5$ of these symmetries are derangements, meaning that they do not fix any sides.

While Theorem 3.2.7 gave us our first way to efficiently compute the expected value of the first letter of a permutation on kn letters with a given number of k-cycles, we can also compute this efficiently with Theorem 3.3.8 by using the formulas for $D(k,n)$ in Theorem 3.3.5. But this is not the only reason that Theorem 3.3.8 is of interest; because of the structure of the formula it provides, this theorem suggests other quantitative and qualitative insights.

Recall that when there are no restrictions on a permutation $\pi \in S_{kn}$, the first letter is equally likely to take on any value, so $\mathbb{E}[\pi(1) \mid \pi \in S_{kn}] = (kn+1)/2$. The first insight given by Theorem 3.3.8 is that the expected value of $\pi(1)$ given some number of k cycles differs from $(kn+1)/2$ by at most $1/2$, because $D(k,N) \geq 1$ for $k \geq 2$. Secondly, since $D(k,N)$ increases as a function of N, the expected value gets closer to $(kn+1)/2$ as the number of k-cycles decreases. Lastly, the numerator of $(-1)^{n-m}$ in the second summand of Equation 3.45 shows that the expected value of the first letter is larger than $(kn+1)/2$ if and only if n and m have the same parity.

3.4 A k-cycle preserving bijection

Motivated by Equation 3.3a, this section describes a family of bijections,

$$\phi_k \colon S_{n-1} \times [n] \to S_n,$$

each of which preserves the number of k-cycles when $k \nmid n$. Of course, there is no map that preserves the number of k-cycles when $k \mid n$. For example, a permutation in S_n consisting entirely of k-cycles contains n/k k-cycles, while a permutation in S_{n-1} can contain at most $n/k - 1$ k-cycles by the pigeonhole principle.

Informally, these maps are defined by writing down a permutation $\sigma \in S_{n-1}$ in *canonical cycle notation*, incrementing all letters in σ that are greater than or equal to $x \in [n]$, inserting x into the rightmost cycle, and then recursively moving letters into or out of subsequent cycles, whenever a k-cycle is turned into a $(k+1)$-cycle or a $(k-1)$-cycle is turned into a k-cycle.

3.4.1 Example of recursive structure

The definition of the map can look complicated, so it's worthwhile to start with an example to give some sense of the overarching idea.

Example 3.4.1. *This example illustrates how the map ϕ_3 inserts I into the permutation $(D76)(E)(F32)(G91C)(K5$ while preserving the number of 3-cycles. The maps ϕ_k and ψ_k are the result of moving letters according to the arrows and are applied from right-to-left. (This example uses the convention that $1 < 2 < \cdots < 9 < A < B < \cdots < N$.)*

$$\begin{aligned}&\phi_3((D76)(E\textvisiblespace)(F\underline{3}2\textvisiblespace)(G\underline{9}1\underline{C})(K\textvisiblespace 5\underline{4})(L\textvisiblespace J\underline{8})(M\textvisiblespace B\textvisiblespace)(N\underline{A}H\textvisiblespace),\underline{I})\\&= (D76)(E3)(F29)(G1)(KC5)(L4J)(M8BA)(NHI)\end{aligned}$$

$$\begin{gathered}\overset{\psi_3}{\curvearrowright}\quad\overset{\psi_3}{\curvearrowright}\quad\overset{\phi_3}{\curvearrowleft}\quad\overset{\phi_3}{\curvearrowleft}\quad\overset{\phi_3}{\curvearrowleft}\quad\overset{\psi_3}{\curvearrowright}\quad\overset{\psi_3}{\curvearrowright}\\ \psi_3((D76)(E\underline{3})(F{\llcorner\!\lrcorner}2\underline{9})(G{\llcorner\!\lrcorner}1{\llcorner\!\lrcorner})(K\underline{C}5{\llcorner\!\lrcorner})(L\underline{4}J{\llcorner\!\lrcorner})(M\underline{8}B\underline{A})(N{\llcorner\!\lrcorner}H\underline{I})){\llcorner\!\lrcorner}\\ = ((D76)(E)(F32)(G91C)(K54)(LJ8)(MB)(NAH),I)\end{gathered}$$

Again, it is worth reemphasizing that the following definitions will follow the convention that permutations are written in canonical cycle notation,

$$\pi = \underbrace{(c_1^{(t)} \cdots c_{\ell_t}^{(t)})}_{c^{(t)}} \cdots \underbrace{(c_1^{(1)} \cdots c_{\ell_1}^{(1)})}_{c^{(1)}},$$

where cycle $c^{(i)} = (c_1^{(i)} \cdots c_{\ell_i}^{(i)})$ has ℓ_i letters. This means that the first letter in each cycle, $c_1^{(i)}$, is the largest letter in that cycle, and that the cycles are ordered in increasing order by first letter when read from right-to-left: $c_1^{(i+1)} < c_1^{(i)}$ for all i.

3.4.2 Formal definition and properties

$$\overset{\phi_3}{\curvearrowright}\quad\overset{\phi_3}{\curvearrowright}\quad\overset{\psi_3}{\curvearrowright}\quad\overset{\psi_3}{\curvearrowright}\quad\overset{\psi_3}{\curvearrowright}\quad\overset{\phi_3}{\curvearrowleft}\quad\overset{\phi_3}{\curvearrowleft}$$

Definition 3.4.2. *Define $\phi_k\colon S_{n-1} \times [n] \mapsto S_n$ recursively as follows:*

$$\phi_k(\emptyset, 1) = (1), \tag{3.52}$$

and for $n > 1$, $\pi \in S_{n-1}$, and $x \in [n]$,

$$\phi_k(\pi,x) = \left\{\begin{array}{lll} c^{(t)}\cdots c^{(1)}(x) & x > c_1^{(1)} & \text{(3.53a)} \\ \phi_k(c^{(t)}\cdots c^{(2)}, c_2^{(1)})(c_1^{(1)}c_3^{(1)}\cdots c_k^{(1)}x) & \ell_1 = k & \text{(3.53b)} \\ \pi'(c_1^{(1)}x'c_2^{(1)}\cdots c_{k-1}^{(1)}x) & \ell_1 = k-1, t > 1 & \text{(3.53c)} \\ c^{(t)}\cdots c^{(2)}(c_1^{(1)}\cdots c_{\ell_1}^{(1)}x) & \textit{otherwise.} & \text{(3.53d)} \end{array}\right.$$

Here, ϕ_k depends on the auxillary function $\psi_k \colon S_n \mapsto S_{n-1} \times [n]$,

$$\psi_k(\pi) = \left\{\begin{array}{lll} \left(c^{(t)}\ldots c^{(2)}, c_1^{(1)}\right) & \ell_1 = 1 & \text{(3.54a)} \\ \left(\phi_k(c^{(t)}\cdots c^{(2)}, a_2^{(1)})(c_1^{(1)}c_3^{(1)}\ldots c_k^{(1)}), c_{k+1}^{(1)}\right) & \ell_1 = k+1 & \text{(3.54b)} \\ \left(\pi'(c_1^{(1)}x'c_2^{(1)}\ldots c_{k-1}^{(1)}), c_k^{(1)}\right) & \ell_1 = k, t > 1 & \text{(3.54c)} \\ \left(c^{(t)}\cdots c^{(2)}(c_1^{(1)}\cdots c_{\ell_1-1}^{(1)}), c_{\ell_1}^{(1)}\right) & \textit{otherwise,} & \text{(3.54d)} \end{array}\right.$$

and in both functions, $(\pi', x') = \psi(c^{(t)}\ldots c^{(2)})$.

Note 3.4.3. *Strictly speaking, ϕ_k and ψ_k have an additional implicit parameter n, which indicates the size of permutation that these functions act on. Since the construction of these functions do not depend on n, this is suppressed in the notation.*

The following theorem motivates this map, and together with Lemma 3.4.7, it implies Equation 3.3a.

Theorem 3.4.4. *If $k \nmid n$, the number of k-cycles of $\pi \in S_{n-1}$ is equal to the number of k-cycles in $\phi_k(\pi,x)$.*

Proof. By construction, the maps ϕ_k and ψ_k change the rightmost cycle into a (different) k-cycle if it was previously a k-cycle, and they change non-k-cycles into non-k-cycles, except for the case where there is one cycle remaining with length $k-1$ (in the case of ϕ) or length k (in the case of ψ). These cases can only be achieved when $k \mid n$, by the following lemma. □

Lemma 3.4.5. *The number of letters in π in (recursive) applications of ϕ_k and ψ_k are of congruent to $n-1 \bmod k$ and $n \bmod k$, respectively. Therefore, the only time that the input to ϕ_k can be a single cycle of length $k-1$ or the input to ψ_k can be a single cycle of length k is when $n \equiv 0 \pmod{k}$.*

Proof. The proof proceeds by induction on the number of recursive iterations of ϕ_k and ψ_k. The base case is clear: on the first application of a map is always $\phi_k \colon S_{n-1} \times [n] \to S_n$, and the input permutation has $n-1$ letters by definition.

Now, either we're finished, or we recurse (Equations 3.53b, 3.53c, 3.54b, or 3.54c), which we look at case-by-case.

Case 1. In Equation 3.53b, the map ϕ_k sets aside k letters from the input, so the number of letters in the recursive input to ϕ_k is also congruent to $n-1 \bmod k$.

Case 2. In Equation 3.53c, the map ϕ_k sets aside $k-1$ letters from the leftmost cycle of the input. Since the number of letters in the original permutation was congruent to $n-1 \bmod k$, the number of letters in the permutation being input to ψ_k is congruent to $n \bmod k$.

Case 3. In Equation 3.54b, the map ψ_k sets aside $k+1$ letters from the leftmost cycle of the input. Since the number of letters in the original permutation was congruent to $n \bmod k$, the number of letters in the permutation being input to ϕ_k is congruent to $n-1 \bmod k$.

Case 4. In Equation 3.54c, the map ψ_k sets aside k letters from the input, so the number of letters in the recursive input to ψ_k is also congruent to $n \bmod k$.

□

The following lemma provides a certain "niceness" property of the map, which allows us to analyze it. In particular, all recursive inputs in both ϕ_k and ψ_k are written in canonical cycle notation.

Lemma 3.4.6. *The output of ϕ_k is in canonical cycle notation.*

Proof. Canonical cycle notation is preserved by construction. In particular, ϕ_k moves the first letter in any cycle, and Equation 3.53a guards against inserting a number into a cycle that is bigger than the largest number already in the cycle. Similarly, ψ_k only moves the first letter in the case of Equation 3.54a, but in this case, the cycle only has one letter, so this is equivalent to deleting the cycle. □

3.4.3 Inverting the bijection

Lemma 3.4.7. *The maps $\phi_k \colon S_{n-1} \times [n] \to S_n$ and $\psi_k \colon S_n \to S_{n-1} \times [n]$ are inverse to one another.*

Proof. To prove this lemma, it suffices to show that $\psi_k \circ \phi_k = \mathrm{id}$ by induction on the number of cycles of π. This will simultaneously prove that $\phi_k \circ \psi_k = \mathrm{id}$, because $S_{n-1} \times [n]$ and S_n, both having $n!$ elements, have the same cardinality.

When π has no cycles, the base case is clear: $\psi_k(\phi_k(\emptyset, x)) = \psi_k((x)) = (\emptyset, x)$.

Now there are five remaining cases to check, corresponding to each of the cases in the definition of $\phi_k(\pi, x)$

Case 1. Assume $x > c_1^{(1)}$, so that $\phi_k(\pi, x)$ is evaluated via Equation 3.53a:

$$\psi_k(\phi_k(\pi, x)) = \psi_k(c^{(t)} \cdots c^{(1)}(x)) \tag{3.55}$$

$$= (c^{(t)} \cdots c^{(1)}, x) \tag{3.56}$$

$$= (\pi, x). \tag{3.57}$$

Case 2. Assume $\ell_1 = k$, so that $\phi_k(\pi, x)$ is evaluated via Equation 3.53b:

$$\psi_k(\phi_k(\pi, x)) = \psi_k(\phi_k(c^{(t)} \cdots c^{(2)}, c_2^{(1)}) \underbrace{(c_1^{(1)} c_3^{(1)} \cdots c_k^{(1)} x)}_{\text{length } k}) \tag{3.58}$$

$$= (\pi'(c_1^{(1)} x' c_3^{(1)} \ldots c_k^{(1)}), x) \tag{3.59}$$

Case 3. Assume $\ell_1 = k-1$ and $t > 1$, so that $\phi_k(\pi,x)$ is evaluated via Equation 3.53c:

$$\psi_k(\phi_k(\pi,x)) = \psi_k(\pi' \underbrace{(c_1^{(1)} x' c_2^{(1)} \cdots c_{k-1}^{(1)} x)}_{\text{length } k+1}) \tag{3.60}$$

where $(\pi',x') = \psi_k(c^{(t)} \ldots c^{(2)})$. Therefore, this simplifies by Equation 3.54c:

$$\begin{aligned} \psi_k(\phi_k(\pi,x)) &= \Big(\phi_k(\pi',x')(c_1^{(1)} \cdots c_{k-1}^{(1)}),x\Big) && (3.61) \\ &= \Big(\underbrace{\phi_k(\psi_k(c^{(t)} \ldots c^{(2)}))}_{c^{(t)} \ldots c^{(2)}} \underbrace{(c_1^{(1)} \cdots c_{k-1}^{(1)})}_{c^{(1)}},x\Big) && (3.62) \\ &= (\pi,x), && (3.63) \end{aligned}$$

because $\phi_k(\psi_k(c^{(t)} \ldots c^{(2)})) = c^{(t)} \ldots c^{(2)}$ by the induction hypothesis on $t-1$ letters.

Case 4. Assume that $x > c_1^{(1)}$ and $\ell_1 \notin \{k-1,k\}$, so that $\phi_k(\pi,x)$ is evaluated via Equation 3.53d:

$$\begin{aligned} \psi_k(\phi_k(\pi,x)) &= \psi_k(c^{(t)} \cdots c^{(2)}(c_1^{(1)} \cdots c_{\ell_1}^{(1)} x)) && (3.64) \\ &= (c^{(t)} \cdots c^{(1)},x) && (3.65) \\ &= (\pi,x). && (3.66) \end{aligned}$$

Case 5. Assume that $\ell_1 = k-1$ and $t = 1$, so that $\phi_k(\pi,x)$ is evaluated via Equation 3.53d:

$$\begin{aligned} \psi_k(\phi_k(\pi,x)) &= \psi_k((c_1^{(1)} \cdots c_{k-1}^{(1)} x)) && (3.67) \\ &= (c^{(1)},x) && (3.68) \\ &= (\pi,x). && (3.69) \end{aligned}$$

□

In this section we constructed a recursively-defined map and its inverse to give a bijective proof that $C_k(n,m) = nC_k(n-1,m)$ when $k \nmid n$. This is a novel, reversible algorithm for inserting a letters into a permutation that preserves the number of k-cycles whenever possible.

3.5 Further directions

In the introduction, we mentioned Conger's paper which analyzed how the number of descents of a permutation affects the expected value of the first letter of the permutation. And similarly in the following sections, we looked at how the number of k-cycles affects the expected value of the first letter of the permutation. This section will principally look at the obvious generalization: given some permutation statistic $\operatorname{stat}\colon S_n \to \mathbb{Z}$, does the map

$$f(n,m) = \mathbb{E}[\pi(i) \mid \pi \in S_n, \operatorname{stat}(\pi) = m] \tag{3.70}$$

have any interesting structure?

But notice that the first letter of a permutation is itself a statistic, so we can play a more general game. Given pairs of statistics $(\operatorname{stat}_1, \operatorname{stat}_2)$, does the map

$$g(n,m) = \mathbb{E}[\operatorname{stat}_1(\pi) \mid \pi \in S_n, \operatorname{stat}_2(\pi) = m] \tag{3.71}$$

have any interesting structure?

3.5.1 FindStat database

The result by Conger gives the expected value of $\pi(1)$ given $\operatorname{des}(\pi)$, and this paper gave the expected value of $\pi(1)$ given the number of k-cycles of π. Of course, it would be interesting to do analogous analysis with other permutations. In particular, the FindStat permutation statistics database [6] contains over 370 different permutation statistics, and many of these appear to have some structure with respect to the expected value of the first letter of a permutation.

3.5.2 Mahonian statistics

In particular, the family of Mahonian statistics may be fruitful to investigate. Below, we have given conjectures about two: the major index and the inversion number. Mahonian statistics are maps $\mathrm{mah}\colon S_n \to \mathbb{N}_{\geq 0}$ that are equidistributed with the inversion number.[7] That is,

$$\#\{w \in S_n : \mathrm{mah}(w) = k\} = \#\{w \in S_n : \mathrm{inv}(w) = k\}.$$

Naturally, all Mahonian statistics share the same generating function:

$$\sum_{\sigma \in S_n} x^{\mathrm{mah}(\sigma)} = [n]_q! = \prod_{i=0}^{n-1} \sum_{j=0}^{i} (q^j).$$

Because the expected value of the first letter is given by the weighted sum of the permutations with $\mathrm{mah}(w) = k$ divided by the number of such permutations, $\mathbb{E}[\pi(1) \,|\, \pi \in S_n, \mathrm{mah}(\pi) = k]$ has a denominator that is (a factor of) $M(n,k)$, the number of permutations of $w \in S_n$ such that $\mathrm{inv}(w) = k$. For fixed k, these satisfy a degree k polynomial for all $n > k$. Notably, in the cases of the major index and the inversion number, the numerators appear to satisfy degree k and degree $k-1$ polynomials respectively.

Conjecture 3.5.1. *For fixed k and n > k, the expected value of the first letter of a permutation with a given number of inversions satisfies a rational function in n given by*

$$\mathbb{E}[\pi(1) \,|\, \pi \in S_n, \mathrm{inv}(\pi) = k] = \frac{M(n+1,k)}{M(n,k)},$$

where $M(n,k)$, as above, is the number of permutations $w \in S_n$ such that $\mathrm{inv}(w) = k$.

Conjecture 3.5.2. *For fixed $k > 0$ and $n \geq k$, $\mathbb{E}[\pi(1) \,|\, \pi \in S_n, \mathrm{maj}(\pi) = k]$ satisfies a rational function in n that is $1/(k+1)$ times the quotient of a monic degree-$(k+1)$ polynomial by a monic degree-k polynomial. Specifically,*

$$\mathbb{E}[\pi(1)\,|\,\pi \in S_n, \mathrm{maj}(\pi) = 1] = \frac{1}{2}\left(\frac{n^2+n-2}{n-1}\right), \tag{3.72}$$

$$\mathbb{E}[\pi(1)\,|\,\pi \in S_n, \mathrm{maj}(\pi) = 2] = \frac{1}{3}\left(\frac{n^3-n-6}{n^2-n-2}\right), \tag{3.73}$$

$$\mathbb{E}[\pi(1)\,|\,\pi \in S_n, \mathrm{maj}(\pi) = 3] = \frac{1}{4}\left(\frac{n^4+6n^3-13n^2-18n}{n^3-7n}\right), \textit{ and} \tag{3.74}$$

$$\mathbb{E}[\pi(1)\,|\,\pi \in S_n, \mathrm{maj}(\pi) = 4] = \frac{1}{5}\left(\frac{n^5+20n^4-45n^3-80n^2-16n}{n^4+2n^3-13n^2-14n}\right). \tag{3.75}$$

Note that the denominator is given by an integer multiple of $M(n,k)$, a degree k polynomial.

3.5.3 An elusive bijection

Let $F_k(n,m)$ be the number of elements of the generalized symmetric group $S(k,n) = (\mathbb{Z}/k\mathbb{Z}) \wr S_n$ with m fixed points, and recall that $C_k(n,m)$ is the number of elements of S_{kn} with m k-cycles. Then for each pair of nonnegative integers (α, β) with $\alpha, \beta \leq n$, then as Lemma 3.3.7 suggests, there exists a bijection of sets

$$C_k(n,\alpha) \times F_k(n,\beta) \to C_k(n,\beta) \times F_k(n,\alpha). \tag{3.76}$$

This bijection has proven to be elusive to construct outside of the special cases where $n = 1$ or $k = 1$. Note that, the map cannot be a group automorphism of $S_{kn} \times S(k,n)$, because the identity of this group is in $C_k(n,0) \times F_k(n,n)$, so it cannot be preserved under this map.

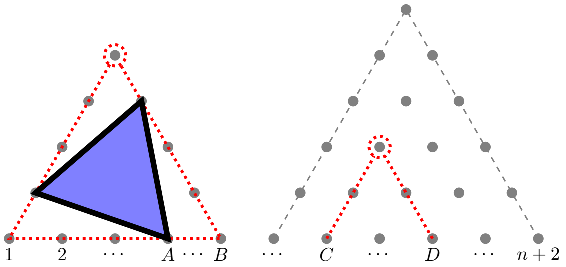
It would be especially interesting if there's a way to use the embedding of $(\mathbb{Z}/k\mathbb{Z}) \wr S_n$ into S_{kn} as the centralizer of an element that is the product of n disjoint k cycles.

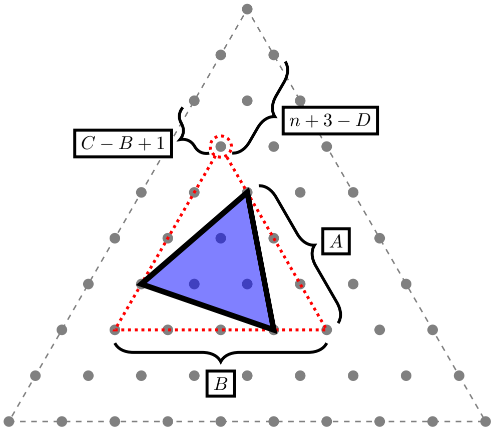
Chapter 4

Interlude: Triangles in Triangles

There are $\binom{n+2}{4}$ equilateral triangles with vertices in a triangular region of the triangular grid with n vertices on each side.

Proof. The following is a bijection without words from a choice of four integers satisfying $1 \le A < B < C < D \le n+2$ to equilateral triangles in the n-vertices-per-side triangular grid.





□

Note 4.0.1 (Formerly the Abstract)**.** *We provide a visual proof of a bijection from* 4*-element subsets of* $\{1,2,\ldots,n+2\}$ *to triangles in the triangular grid where the orientation of the triangle is given by the smallest element of the subset, the size of its bounding triangle is by the second smallest element, and the position of its bounding triangle is given by the two largest elements. In this example,* $n = 10$*,* $A = 4$*,* $B = 5$*,* $C = 6$*, and* $D = 10$*.*

Chapter 5

Deranking Menage

TODO

5.1 TODO

1. Introduction

2. We can also *rank* a given permutation

3. Define **derived** complementary board B^c_α?

4. If we do a cyclic rotation of the rows of a chessboard, we get essentially the same thing.

5. Move code to Appendix.

6. Define B_α and $\overline{B}^c_\alpha$.

7. Do we want to talk about parking functions?

8. Is it worthwhile to discuss prefix functions for compositions, etc.?

5.2 Overview and History

In January 2020, Richard Arratia sent out an email announcing a talk he was going to give on de-ranking derangements.

By January 2021, he announced a \$100 prize for solving the analogous problem with ménage permutations. I solved that too.

Richard was interested in a more general question, which I found contagious: Given some family of combinatorial objects that can be quickly counted (say unlabelled simple graphs on n vertices) and some total ordering on them, when is it possible to **derank** the collection in some computationally efficient way?

Of course, we can usually create an algorithm to give the i-th object without simply enumerating all of the objects explicitly? We want to "jump in" to a specific place on the list. Another interesting question: what if you get to supply both the total order and the deranking algorithm?

In this chapter we're going to explore that idea. We're going to show a general theory that allows us to de-rank permutations in lexicographic order, derangements in lexicographic order, partitions and compositions of n in lexicographic order, labeled trees by lexicographic order of Prüfer code, Lyndon words [8] (de Bruijn Sequences?), Dyck path in lexicographic order?

5.3 Overarching Theory (count with prefixes)

If we can efficiently count how many objects in $[n]^k$ start with a given prefix (in $O(T(n,k))$ time), then we can just walk down the possible letters until we get to the right spot ($O(nkT(n,k))$).

5.3.1 Counting Words With a Given Prefix

TODO: We can reduce this problem to counting words with a given prefix.

Lemma 5.3.1. *Let $\mathscr{W}_k \subseteq [n]^k$ be an ordered collection of words of length k on an alphabet of size n, and denote the set of nonempty candidate prefixes by $\mathscr{P}_k = [n] \cup [n]^2 \cup \cdots \cup [n]^k$ Then given a function $\#\text{prefix}\colon \mathscr{P}_k \to \mathbb{N}$ that counts the number of words that begin with a given prefix, the i-th word in $\mathscr{W}_k$ when written in lexicographic order is*

$$\text{derank}_i((1),0)$$

which can be computed explictly with nk or fewer recursive calls:

$$\operatorname{derank}_i(\alpha, b) = \begin{cases} \alpha & i \in (b, b + \#\text{prefix}(\alpha)] \textit{ and } \alpha \in \mathscr{W}_k \\ \operatorname{derank}_i(\alpha', b) & i \in (b, b + \#\text{prefix}(\alpha)] \textit{ and } \operatorname{len}(\alpha) < k \\ \operatorname{derank}_i(\alpha'', b + \#\text{prefix}(\alpha)) & \textit{otherwise}, \end{cases} \tag{5.1}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$, $\alpha' = (\alpha_1, \alpha_2, \ldots, \alpha_\ell, 1)$, $\alpha'' = (\alpha_1, \alpha_2, \ldots, \alpha_{k-1}, \alpha_\ell + 1)$, and b denotes the number of words in $\mathscr{W}_k$ that occur strictly ***before*** *α.*

Proof. TODO (sketch) The second line appends a letter, which can happen at most n times. The third line increments the last letter, which can happen at most k times per position. □

By choosing the appropriate counting function #prefix, this translates the problem from the domain of deranking objects to the domain of counting the number of objects with a given prefix. This technique works when we can write our objects as a word in $[n]^k$, and we order the objects by the lexicographic order of the words. In the case that our objects cannot be written as words, or we are interested in an order other than lexicographic order, a different technique must be used.

5.3.2 Ranking words

TODO: We can also take a word $w \in \mathscr{W}_k$ and quickly determine its rank.

5.3.3 Basic Notions of Rook Theory

In the case of deranking derangements and permutations, it is useful to use ideas from Rook Theory. Rook Theory was introduced by Kaplansky [9] Riordan [10] in their 1946 paper *The Problem of the Rooks and its Applications*. In it, they discuss problems of restricted permutations in the language of rooks placed on a chessboard. We begin by introducing some preliminary ideas in this theory.

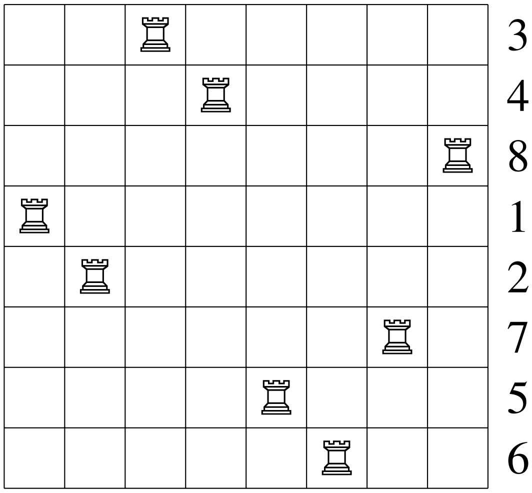


Figure 5.1: An illustration of the rook placement corresponding to the permutation $34812756 \in S_8$. A rook is placed in square $(i, \pi(i))$ for each i.

Definition 5.3.2. *A board B is a subset of $[n] \times [n]$ which represents the squares of a $n \times n$ chessboard that rooks are allowed to be placed on. Every board B has a complementary board $B^c = ([n] \times [n]) \setminus B$, which consists of all of the squares of B that a rook cannot be placed on.*

To each board, we can associate a generating polynomial that keeps track of the number of ways to place a given number of rooks on the valid squares in such a way that no two rooks are in the same row or column.

Definition 5.3.3. *The rook polynomial associated with a board B,*

$$p_B(x) = r_0 + r_1 x + r_2 x^2 + \cdots + r_n x^n,$$

is a generating polynomial where r_k denotes the number of k-element subsets of B such that no two elements share an x-coordinate or a y-coordinate.

In the context of permutations, we're typically interested in r_n, the number of ways to place n rooks on a restricted $n \times n$ board. However, it turns out that a naive application of the techniques from rook theory do not immediately allow us to count the number of restricted permutations with a given prefix. Computing the number of such permutations is known to be computationally hard for a board with arbitrary restrictions. We can see this by encoding a board B as a $(0,1)$-matrix and computing the matrix permanent. (In fact, Shevelev [11] claims that “the theory of enumerating the permutations with restricted positions stimulated the development of the theory of the permanent.”)

Lemma 5.3.4. *Let $M_B = \{a_{ij}\}$ be an $n \times n$ matrix where*

$$a_{ij} = \begin{cases} 1 & (i,j) \in B \\ 0 & (i,j) \notin B \end{cases}.$$

Then the coefficient of x^n in $p_B(x)$ is given by the matrix permanent

$$\mathrm{perm}(M_B) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}.$$

Now is the perfect time to recall Valiant’s Theorem.

Theorem 5.3.5 (Valiant’s Theorem [12])**.** *Computing the permanent of a (0,1)-matrix is #P-complete.*

Corollary 5.3.6. *Computing the number of rook placements on an arbitrary $n \times n$ board is #P-hard.*

Therefore, in order to compute the number of permutations, we must exploit some additional structure of the restrictions.

5.3.4 Techniques of Rook Theory

Rook polynomials can be computed recursively. The base case is that for an empty board $B = \emptyset$, the corresponding rook polynomial is $p_\emptyset(x) = 1$, because there is one way to place no rooks, and no way to place one or more rooks.

Lemma 5.3.7 ([10])**.** *Given a board, B, then for any square $(x,y) \in B$, we can define the resulting boards if we include or exclude the square respectively*

$$B_i = \{(x',y') \in B : x \neq x' \text{ and } y \neq y'\} \tag{5.2}$$

$$B_e = B \setminus (x,y). \tag{5.3}$$

Then we can write the rook polynomial for B in terms of this decomposition.

$$p_B(x) = x p_{B_i}(x) + p_{B_e}(x).$$

If we want to compute a rook polynomial using this construction, we can end up adding up lots of smaller rook polynomials—a number that is exponential in the size of B. However, when the number of squares in B^c is small in some sense, it can be easier to compute the rook polynomial p_{B^c} and use the principle of inclusion/exclusion on it's coefficients to determine the rook polynomial for the original board, B.

In the case of derangements and ménage permutations, this is the strategy we'll use. Start by finding the resulting board from a given prefix, find the rook polynomial of the complementary board, and use the principle of inclusion/exclusion to determine the number of ways to place rooks in the resulting board.

5.4 Deranking Derangements

In January 2020, Richard Arratia sent out an email proposing a seminar talk. The title describes the first "\$100 problem":

\$100 Problem. *"For* 100 *dollars, what is the* 500 *quadrillion-th derangement on* $n = 20$*?"*

\$100 Answer. *The computer program in Appendix TODO computed the answer in less than ten milliseconds. When written as words in lexicographic order, the derangement in S_{20} with rank 5×10^{17} is*

$$12\ 14\ 2\ 9\ 13\ 20\ 6\ 3\ 1\ 17\ 5\ 11\ 19\ 15\ 10\ 18\ 8\ 7\ 4\ 16.$$

Arratia's question focused on deranking derangements where the rank was based on the total ordering that comes from writing the permutations as words in lexicographic order. Other authors have looked at deranking derangements based on other total orderings. In particular, Mikawa and Tanaka [13] give an algorithm to rank/unrank derangements with respect to *lexicographic ordering in cycle notation*.

In this section we will develop an algorithm for ranking and deranking with respect to their lexicographic ordering as words. The technique that we use will broadly be re-used in the next section. It is worthwhile to begin by recalling the definition of a derangement.

Definition 5.4.1. *A derangement is a permutation $\pi \in S_n$ such that π has no fixed points. That is, the set of derangements is*

$$\{\pi \in S_n : \pi(i) \neq i \;\forall i \in [n]\}.$$

5.4.1 The complementary board.

In order to compute the number of derangements with a given prefix, it is useful to look at the board that results after placing k rooks according to these positions, as illustrated in Figure 5.2.

Definition 5.4.2. *If B is an $n \times n$ board, and $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ is a valid prefix of length ℓ, then derived complementary board of B from α, denoted B^c_α, is B with the appropriate rows and columns removed and reindexed in such a way that $B^c_\alpha \subseteq [n-\ell] \times [n-\ell]$.*

Lemma 5.4.3. *Given a valid ℓ letter prefix $(\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ of a word on n letters, the number of squares in the resulting complementary board is*

$$|B^c_\alpha| = n - \ell - |\{\ell+1, \ell+2, \ldots, n\} \cap \{\alpha_1, \alpha_2, \ldots, \alpha_\ell\}|,$$

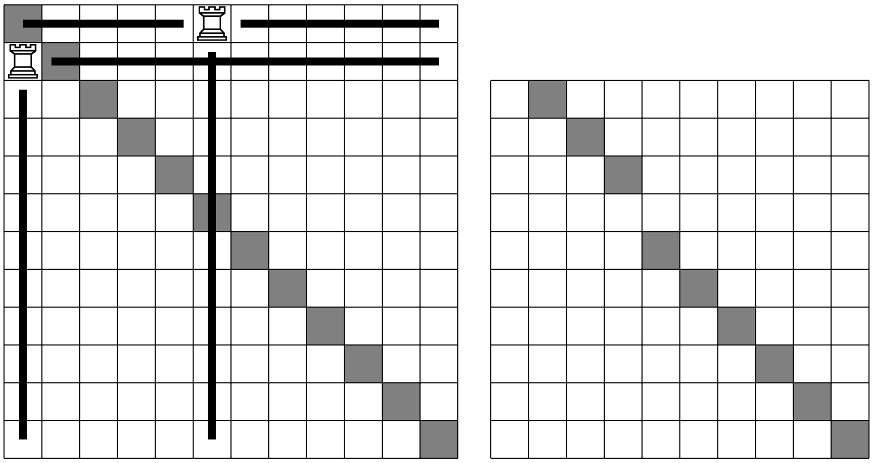


Figure 5.2: An example of a prefix $\alpha = (6,1)$, and the board that results from deleting the first $\ell = 2$ rows and columns 6 and 1. The derived complementary board of B from α is $B^c_\alpha = \{(1,2),(2,3),(3,4),(5,5),\ldots,(10,10)\}$.

and no two of these squares are in the same row or column.

Proof. TODO □

5.4.2 Derangements with a given prefix

Now that we have a way of quickly computing $|B^c_\alpha|$, we can compute the number of ways to place j rooks on the complementary board. We can use this to compute the number of derangements that begin with the prefix α.

Lemma 5.4.4. *The rook polynomial for the complementary board B^c_α is*

$$p_{B^c_\alpha}(x) = \sum_{j=0}^{|B^c_\alpha|} \binom{|B^c_\alpha|}{j} x^j. \tag{5.4}$$

Proof. No two squares in B^c (and thus B^c_α) are in the same row or column. Thus the number of ways to place j rooks is equivalent to selecting j cells from $|B^c_\alpha|$. □

Now we introduce a lemma of Stanley [14] to compute the number of TODO from a complementary board.

Lemma 5.4.5 ([14]). *The number of ways, N_0, of placing n nonattacking rooks on a board $B \subseteq [n] \times [n]$ is given by*

$$N_0 = \sum_{k=0}^{n} (-1)^k r_k (n-k)!,$$

where $P_{B^c}(x) = \sum_{k=0}^{n} r_k x^k$.

Corollary 5.4.6. *The number of derangements with prefix $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_\ell)$ is given by*

$$\#\text{prefix}(\alpha) = \sum_{j=0}^{|B^c_\alpha|} (-1)^j \binom{|B^c_\alpha|}{j} (n-\ell-j)!,$$

which is A047920$(n-\ell, |B^c_\alpha|)$ *in the On-Line Encyclopedia of Integer Sequences [2].*

Example 5.4.7. *For example, for $N = 14$, we wish to count the number of derangements that start with the prefix* $6\,1$. *Since the prefix has two letters, $p = 2$ and $n = 14 - 2 = 12$. The only crossed-out cell that is deleted by the prefix in the remaining board is the cell that was in position* 6: *in particular, $\{3, 4, \dots, 14\} \cap \{6, 1\} = 6$. Thus $k = 12 - 1 = 11$. Thus there are $A047920(12, 11) =$* $190\,899\,411$ *derangements that start with* $6\,1$.

5.5 Deranking Ménage Permutations

A Ménage permutation comes from the *problème des ménages*. Here we will define it as

Definition 5.5.1. *A ménage permutation is a permutation $\pi \in S_n$ such that for all $i \in [n]$, $\pi(i) \neq i$ and $\pi(i) + 1 \not\equiv i \bmod n$.*

We can use the prefix to get a new board, which is block diagonal (whenever the prefix is non-empty), if we know the number of cells in each block, we can compute the number of valid boards. This gives us the number of ménage permutations with a given prefix.

Prefix ⇒ grouped columns ⇒ partition/multiset ⇒ complementary polynomial ⇒ count

α (prefix)	#prefix(α)	index range	$\lvert B_{\alpha}^{c}\rvert$	$\mathrm{derank}_i(\alpha,\ell)$
1	0	$(0,0]$	–	$\mathrm{derank}_{1000}(1,0)$
2	2119	$(0,2119]$	6	$\mathrm{derank}_{1000}(2,0)$
21	265	$(0,265]$	6	$\mathrm{derank}_{1000}(21,0)$
22	0	$(265,265]$	–	$\mathrm{derank}_{1000}(22,265)$
23	309	$(265,574]$	5	$\mathrm{derank}_{1000}(23,265)$
24	309	$(574,883]$	5	$\mathrm{derank}_{1000}(24,574)$
25	309	$(883,1192]$	5	$\mathrm{derank}_{1000}(25,883)$
251	53	$(883,936]$	4	$\mathrm{derank}_{1000}(251,883)$
253	0	$(936,936]$	–	$\mathrm{derank}_{1000}(253,936)$
254	64	$(936,1000]$	3	$\mathrm{derank}_{1000}(254,936)$
2541	11	$(936,947]$	3	$\mathrm{derank}_{1000}(2541,936)$
2543	11	$(947,958]$	3	$\mathrm{derank}_{1000}(2543,947)$
2546	14	$(958,972]$	2	$\mathrm{derank}_{1000}(2546,958)$
2547	14	$(972,986]$	2	$\mathrm{derank}_{1000}(2547,972)$
2548	14	$(986,1000]$	2	$\mathrm{derank}_{1000}(2548,986)$
25481	3	$(986,989]$	2	$\mathrm{derank}_{1000}(25481,986)$
25483	3	$(989,992]$	2	$\mathrm{derank}_{1000}(25483,989)$
25486	4	$(992,996]$	1	$\mathrm{derank}_{1000}(25486,992)$
25487	4	$(996,1000]$	1	$\mathrm{derank}_{1000}(25487,996)$
254871	2	$(996,998]$	0	$\mathrm{derank}_{1000}(254871,996)$
254873	2	$(998,1000]$	0	$\mathrm{derank}_{1000}(254873,998)$
2548731	1	$(998,999]$	0	$\mathrm{derank}_{1000}(2548731,998)$
2548736	1	$(999,1000]$	0	$\mathrm{derank}_{1000}(2548736,999)$
25487361	1	$(999,1000]$	0	$\mathrm{derank}_{1000}(25487361,999)$

Figure 5.3: There are $A000166(8) = 14833$ derangements on 8 letters. This algorithm finds the derangement at index 1000.

5.5.1 Block diagonal decomposition

When we look at Figure TODO, it appears that placing rooks according to a prefix results in a derived complementary board where the squares can be grouped into sub-boards that don't share any rows or columns. We will see that this property holds more generally, and we can exploit this in order to describe the number of ménage permutations with a given prefix.

It is useful to begin by formalizing this notion of grouping squares.

Definition 5.5.2. *Two boards B and B' are called* ***disjoint*** *if no squares of B are in the same row or column as any square in B'.*

The reason that we care about decomposing a board into disjoint parts is because that perspective allows us to factor the rook polynomial.

Lemma 5.5.3 ([9])**.** *If B can be partitioned into disjoint boards $b_1, b_2, \dots, b_m$, then the rook polynomial of B is the product of the rook polynomials of the b_is*

$$p_B(x) = \prod_{i=1}^{m} p_{b_i}(x).$$

The key insight is that after placing rooks in valid positions in the top $1 \leq k \leq n-1$ rows, we get block-diagonal boards, with three possible shapes, shown in Figure 5.4.

Lemma 5.5.4. *For $\ell \geq 1$, and prefix $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_\ell)$ the derived complementary board B^c_α can be partitioned into boards of one of three shapes.*

1. *(TODO Figure 5.4, left)*

2. *(TODO Figure 5.4, middle)*

3. *(TODO Figure 5.4, right)*

Proof. The proof proceeds by induction. Base case: because of the ménage restriction, $\pi(1) \in \{2, 3, \dots, n-1\}$, and so the resulting board is split into a part of size $2\pi(1)-3$ and $2n-2\pi(1)-1$ parts respectively. Inductive step: TODO □

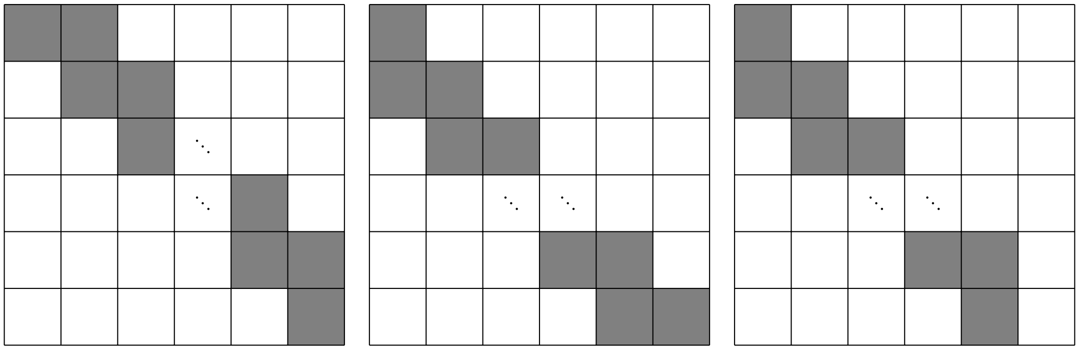


Figure 5.4: Three $n \times n$ blocks, two with $2n-1$ crossed-out cells and one with $2n-2$ crossed-out cells.

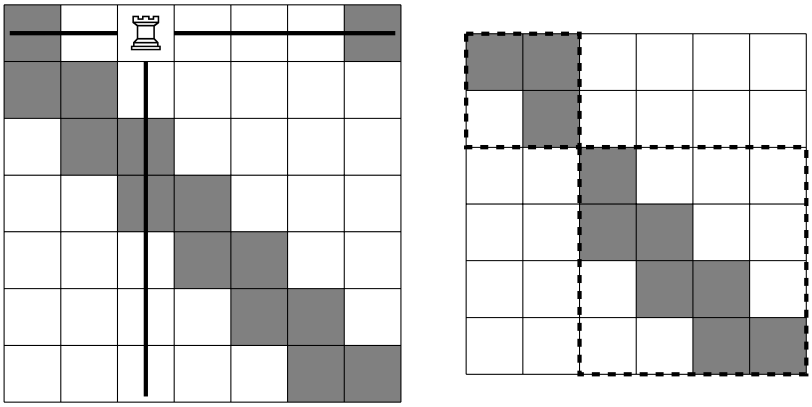


Figure 5.5: The first chessboard shows a placement of a rook at position 3, the second shows the remaining squares, and the third shows a permutation of the rows to put the board into a block-diagonal form.

5.5.2 Rook polynomials of blocks

Recall that the goal of partitioning B into disjoint boards $b_1, b_2, \ldots, b_m$ is so that we can factor $p_B(x)$ in terms of $p_{b_i}(x)$. Of course, this is only helpful if we can describe $p_{b_i}(x)$, which is the goal of this section. Thankfully, the rook polynomial of each b_i will turn out to depend only on the number of squares, $|b_i|$, which can be computed easily because of its structure.

We will begin by defining a family of polynomials that, suggestively, will turn out to be the rook polynomials that we are looking for. This family is nearly described by OEIS sequence A011973 [2].

Definition 5.5.5. *For $j \geq 0$, the jth* ***Fibonacci polynomial*** *$F_j(x)$ is defined recursively as:*

$$F_0(x) = 1 \tag{5.5}$$

$$F_1(x) = 1 + x \tag{5.6}$$

$$F_n(x) = F_{n-1}(x) + xF_{n-2}(x). \tag{5.7}$$

Lemma 5.5.6. *Given a board B that consists of a single block with k crossed out cells, it's complementary board B^c has rook polynomial $p_{B^c}(x) = F_{k+1}(x)$.*

Proof. We will recall Lemma 5.3.7, and proceed by induction on the upper-left square.

TODO (See Figure 5.4, boards A, B, C)

Base case: If we have a board of type C and size 0, it has a rook polynomial of 1. If we have a board of type A (or B) and size 1, it has a rook polynomial of $1+x$.

Suppose our inductive hypothesis holds for boards with up to s squares. Then

1. B_i for A_{2n-1} is equal to A_{2n-3}. B_e is C_{2n-2}.

2. B_i for B_{2n-1} is equal to B_{2n-3}. B_e is a flip of C_{2n-2} along antidiagonal.

3. B_i for C_{2n-2} is equal to C_{2n-4}. B_e is A_{2n-3} along antidiagonal.

□

5.5.3 Prefix to blocks

Here's the idea: we group the uncrossed columns.

Lemma 5.5.7. *Given a prefix $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_\ell)$ and $i \notin \alpha$, the number of cells of B^c in column i that do not have a first coordinate in $[\ell]$ is given by the rule:*

$$c_i = \begin{cases} 0 & i < \ell \\ 1 & i = k \text{ or } i = n \\ 2 & \ell < i < n \end{cases} \tag{5.8}$$

Proof. TODO: This almost follows from the description? □

Now we can put these column counts together based on the continuous blocks.

Lemma 5.5.8. *TODO: Partition $[n] \setminus \alpha$ into contiguous parts. This naturally partitions B^c_α into disjoint boards. The size of these boards is $\sum_{x \in part} c_x$. (this is what I've been calling our "composition")*

5.5.4 Complementary polynomials to ménage permutations with a given prefix

Recap: We've taken a prefix, used it to find contiguous regions, used these to find disjoint subboards related to B^c_α, whose rook polynomials we know. Now it's time to take these to count our number of ménage permutations with the aforementioned prefix.

Lemma 5.5.9. *Given a board B^c_α that is partitioned into disjoint boards $b_1, b_2, \dots, b_m$, the rook polynomial of B^c_α is*

$$p_{B^c_\alpha}(x) = \prod_{i=1}^{m} F_{b_i}(x).$$

Proof. This follows directly from Lemma (TODO: rook polynomials of blocks) and Lemma (TODO: product of blocks is whole thing). □

Now that we know $p_{B_{\alpha}^{c}}$, we can use Lemma (TODO: Complementary to original) to determine how many ménage permutations there are with a given prefix. Because of Lemma (TODO: all we need is the prefix to derank), we have an algorithm to derank.

5.5.5 Proof of concept (The $100 answer!)

$100 Problem. *For $n = 20$ there are $A000179(20) = 312\,400\,218\,671\,253\,762 > 3.1 \cdot 10^{17}$ ménage permutations. Determine the 10^{17}-th such permutation when listed in lexicographic order.*

$100 Answer. *The desired permutation is*

$$7\ 16\ 19\ 12\ 2\ 8\ 15\ 1\ 18\ 14\ 3\ 9\ 20\ 10\ 5\ 17\ 13\ 4\ 11\ 6. \tag{5.9}$$

Example 5.5.10. *Illustrating this particular example is too big to be of much interest, so here's a smaller example. There are $A000179(8) = 4738$ ménage permutations on 8 letters. We'll use this algorithm to find the one at index 1000.*

5.6 Generalizations and Open Questions

5.6.1 Other restricted permutations

Doron Zeilberger considers a more general family of restricted permutations.

Definition 5.6.1 ([15])**.** *Let $S \subset \mathbb{Z}$, then a S-avoiding permutation is a permutation $\pi \in S_n$ such that*

$$\pi(i) - i - s \not\equiv 0 \bmod n \textit{ for all } i \in [n] \textit{ and } s \in S.$$

Example 5.6.2. *Ordinary permutations are $\emptyset$-avoiding permutations, derangements are $\{0\}$-avoiding permutations, and we've defined menagé permutations as $\{-1, 0\}$-avoiding permutations.*

The results in this paper generalize pretty easily to $\{i, i+1\}$-avoiding permutations for all i.

prefix	starting with prefix	index range	composition	derank$_i(\alpha,\ell)$
1	0	$(0,0]$	–	derank$_{1000}(1,0)$
2	787	$(0,787]$	$(1,11)$	derank$_{1000}(2,0)$
3	791	$(787,1578]$	$(3,9)$	derank$_{1000}(3,787)$
31	0	$(787,787]$	–	derank$_{1000}(31,787)$
32	0	$(787,787]$	–	derank$_{1000}(32,787)$
33	0	$(787,787]$	–	derank$_{1000}(33,787)$
34	159	$(787,946]$	$(1,7)$	derank$_{1000}(34,787)$
35	166	$(946,1112]$	$(1,2,5)$	derank$_{1000}(35,946)$
351	24	$(946,970]$	$(0,2,5)$	derank$_{1000}(351,946)$
$\cdots$	0	$(970,970]$	–	
354	34	$(970,1004]$	$(0,5)$	derank$_{1000}(354,970)$
3541	5	$(970,975]$	$(0,5)$	derank$_{1000}(3541,970)$
3542	5	$(975,980]$	$(0,5)$	derank$_{1000}(3542,975)$
$\cdots$	0	$(980,980]$	–	
3546	8	$(980,988]$	$(0,3)$	derank$_{1000}(3546,980)$
3547	10	$(988,998]$	$(0,2,1)$	derank$_{1000}(3547,988)$
3548	6	$(998,1004]$	$(0,4)$	derank$_{1000}(3548,998)$
35481	1	$(998,999]$	$(0,4)$	derank$_{1000}(35481,998)$
35482	1	$(999,1000]$	$(0,4)$	derank$_{1000}(35482,999)$
354821	0	$(999,999]$	(3)	derank$_{1000}(354821,999)$
$\cdots$	0	$(999,999]$	–	
354827	1	$(999,1000]$	$(0,1)$	derank$_{1000}(354827,999)$
3548271	1	$(999,1000]$	(0)	derank$_{1000}(3548271,999)$
35482716	1	$(999,1000]$	$()$	derank$_{1000}(35482716,999)$

5.6.2 Observation about Lyndon Words after? a given prefix

Definition 5.6.3. *A Lyndon word is a string that is the unique minimum with respect to all of its rotations.*

Example 5.6.4. 00101 *is a Lyndon word because* $00101 = \min\{00101, 01010, 10100, 01001, 10010\}$ *is the unique minimum of all of its rotations.*

011011 *is not a Lyndon word because while* $011011 = \min\{011011, 110110, 101101, 011011, 110110, 101101\}$ *it is not the* ***unique*** *minimum.*

Conjecture 5.6.5. *Let* $\mathscr{E}^{-1}$ *denote the inverse Euler transform. Then the number of length* $n+1$ *Lyndon words that start with a prefix* α *follows a "simple" linear recurrence for sufficiently large* n*.*

Chapter 6

Conclusion and ongoing work

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

References

1. Conger, M. A refinement of the Eulerian numbers, and the joint distribution of $\pi(1)$ and $\mathrm{Des}(\pi)$ in S_n. *Ars Combinatoria* **95** (2010).
2. Inc., O. F. *The On-Line Encyclopedia of Integer Sequences* 2021.
3. V. Goncharov. Du domaine de l'analyse combinatoire. *Izv. Akad. Nauk SSSR Ser. Mat.* **8,** 3–48 (1 1944).
4. Arratia, R. & Tavare, S. The Cycle Structure of Random Permutations. *The Annals of Probability* **20,** 1567–1591 (1992).
5. Assaf, S. H. Cyclic Derangements. *The Electronic Journal of Combinatorics* **17** (2010).
6. Rubey, M., Stump, C., *et al. FindStat - The combinatorial statistics database* `http://www.FindStat.org`. Accessed: May 10, 2022.
7. Foata, D. *Distributions Euleriennes et Mahoniennes sur le Groupe des Permutations* in *Higher Combinatorics* (ed Aigner, M.) (Springer Netherlands, Dordrecht, 1977), 27–49.
8. Kociumaka, T., Radoszewski, J. & Rytter, W. *Computing k-th Lyndon Word and Decoding Lexicographically Minimal de Bruijn Sequence* in *Combinatorial Pattern Matching* (Springer International Publishing, 2014), 202–211.
9. Kaplansky, I. & Riordan, J. The problem of the rooks and its applications. *Duke Mathematical Journal* **13,** 259–268. doi:`10.1215/S0012-7094-46-01324-5` (1946).
10. Riordan, J. *An Introduction to Combinatorial Analysis* (Princeton University Press, USA, 1980).
11. Shevelev, V. S. Some problems of the theory of enumerating the permutations with restricted positions. *Journal of Soviet Mathematics* **61,** 2272–2317. doi:`10.1007/BF01104103` (1992).
12. Valiant, L. The complexity of computing the permanent. *Theoretical Computer Science* **8,** 189–201. doi:`https://doi.org/10.1016/0304-3975(79)90044-6` (1979).
13. Mikawa, K. & Tanaka, K. Lexicographic ranking and unranking of derangements in cycle notation. *Discret. Appl. Math.* **166,** 164–169 (2014).
14. Stanley, R. P. *Enumerative Combinatorics: Volume 1* 2nd (Cambridge University Press, USA, 2011).
15. Zeilberger, D. Automatic Enumeration of Generalized Ménage Numbers. *Séminaire Lotharingien de Combinatoire* **71** (2014).

Appendices

A A Long Proof

A.1 Haskell Algorithm for Ménage

```
import Helpers.Factorials (factorial)
import Data.List (sort, nub)

type Prefix = [Int]
type PolynomialCoefficients = [Integer]
type PrefixCount = Prefix -> Integer

rookCount :: Int -> Integer -> Prefix
rookCount n = derank n n (rookPrefixCount n)

-- derank from alphabet of size n with k letters
-- and a way of counting the number of words with a given prefix
derank :: Int ->                   -- Alphabet of n letters
          Int ->                   -- Words of length k
          (Prefix -> Integer) ->   -- #prefix function
          Integer ->               -- Derank at targetIndex
          Prefix                   -- Word at rank targetIndex
```

```
  derank n k prefixCounter targetIndex = recurse (0, 0) 1 [] where
    recurse :: (Integer, Integer) -> -- index range with given prefix (a, b
               Int ->                -- candidate for current letter
               Prefix ->             -- established prefix
               Prefix                -- word at index
    recurse (a, b) c prefix
      | c > n                               = error ”Out of range!”
      | length prefix == k                  = prefix
      | a < targetIndex && targetIndex <= b = recurse (a, b') 1 (prefix
++ [c])
      | otherwise                           = recurse (b, b'') (c + 1) prefi
      b'     = a + prefixCounter (prefix ++ [c, 1])
      b''    = b + prefixCounter (prefix  ++ [c + 1])

  -- Assumes prefix is valid; no duplicate values or illegal positions.
  -- If n = 9 and the prefix is [3, 8, 7]
  -- This should return [[1,2],[4,5,6],[9]]
  getColumnGroups :: Int -> Prefix -> [[Int]]
  getColumnGroups n prefix = filter (not . null) $ columnGroups where
    cols = 0 : (sort prefix) ++ [n+1]
    columnGroups = zipWith (\a b ->  [a+1..b-1]) cols (tail cols)

  getComposition :: Int -> Prefix -> [Int]
  getComposition n prefix = map (sum . map cellsInColumn) columnGroups wher
    columnGroups = getColumnGroups n prefix
    k = length prefix
    cellsInColumn c
      | c < k      = 0
      | c == k     = 1
      | c == n     = 1
      | otherwise  = 2

  fibonacciPolynomial :: Int -> PolynomialCoefficients
  fibonacciPolynomial = (!!) fibonacciPolynomials where
    fibonacciPolynomials = [1] : [1] : recurse [1] [1] where
      recurse f g = h : recurse g h where
        h = ([0,1] .*. f) .+. g

  complementaryRookPolynomial :: Int -> Prefix -> PolynomialCoefficients
  complementaryRookPolynomial n prefix = foldr (.*.) [1] blockPolynomials w
    blockPolynomials = map (\i -> fibonacciPolynomial (i + 1)) $ getComposi

  invalidPrefix :: Int -> Prefix -> Bool
  invalidPrefix n prefix = containsDuplicates || invalidPosition where
    containsDuplicates = prefix /= (nub prefix)
    invalidPosition = any inRestrictedPosition $ zip [0..] prefix where
      inRestrictedPosition (i, x) = (x `mod` n == i) || (x == i + 1)

  rookPrefixCount :: Int -> Prefix -> Integer
  rookPrefixCount n prefix
    | invalidPrefix n prefix = 0
    | otherwise              = recurse 0 crp where
    n' = fromIntegral (n - length prefix)
    crp = complementaryRookPolynomial n prefix
```

```
    recurse k (c:cs) = (-1)^k * c * factorial (n'-k) + recurse (k+1) cs
    recurse _ [] = 0

  -- The polynomial a + bx + cx^2 ... is represented as
  -- [a, b, c, ...]
  -- These are helper functions for adding and multiplying polynomials
  (.+.) :: PolynomialCoefficients -> PolynomialCoefficients -> PolynomialC
  (.+.) p1 [] = p1
  (.+.) [] p2 = p2
  (.+.) (a:p1) (b:p2) = (a + b) : (p1 .+. p2)

  (.*.) :: PolynomialCoefficients -> PolynomialCoefficients -> PolynomialC
  (.*.) p1 [] = []
  (.*.) [] p2 = []
  (.*.) p1 p2 = foldr1 (.+.) termwiseProduct where
    termwiseProduct = map (\(i,x) -> replicate i 0 ++ map (*x) p2) $ zip [0
```

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.