Permutations, Statistics, and Switches

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

Table of Contents

List of Tables

List of Figures

Chapter 1

Generalized Spinning Switches

In this chapter, we explore puzzles about switches on the corners of a spinning table. Such puzzles have been written about and generalized since they were first popularized by Martin Gardner. In this chapter, we provide perhaps the fullest generalization yet, modeling both the switches and the spinning table as a wreath product of two arbitrary finite groups. We classify large families of wreath products depending on whether or not they correspond to a solvable puzzle, completely resolving the problem for p-groups, and providing novel examples for other families of groups, including two interchangeable copies of the monster group M. Lastly, we provide a number of open questions and conjectures, and provide other suggestions of how to generalize some of these ideas further.

1.1 Overview and Preliminaries

The paper is organized into six sections. This section, Section 1.1 provides a brief history of this genre of puzzles and introduces some of the first approaches to generalizing the puzzle further. Section 1.2 models these generalizations in the context of the wreath product, and formalizes the notation of puzzles being solvable. Section 1.3 explores situations where the puzzle does not have a winning strategy, and provides reductions that allow us to prove that entire families of puzzles are not solvable. Section 1.4 constructs a strategy for switches that behave like p-groups, and gives us ways of building strategies from smaller parts. Section 1.5 provides novel examples of puzzles that do not behave like p-groups, but still have winning strategies. Lastly, Section 1.6 provides further generalizations, and contains dozens of conjectures, open questions, and further directions.

1.1.1 History

Spinning Switches puzzles are a family of closely related puzzles that were first popularized by Martin Gardner in the form of pint glasses on a lazy susan. In the February 1979 edition of his column "Mathematical Games." [1] Gardner writes that he learned of the puzzle from Robert Tappay of Toronto who "believes it comes from the U.S.S.R.," a history that is not especially forthcoming.

My preferred version of the puzzle 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. [2]

(Winkler's version will be a working example in many parts of the paper, so it is worth keeping in mind. An illustration can be found in Figure 1.1.)

Over the last three decades, various authors have consider generalizations of this puzzle. Here, we build on those results and go further. The first place authors looked to generalize was suggested by Gardner himself. In his March 1979 column, he provided the answer to the original puzzle and wrote

> 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. [3]

In 1993, Bar Yehuda, Etzion, and Moran [4]. took on the challenge and developed a theory of the spinning switches puzzle where the switches behave like roulettes with a single “on” state. In this chapter we take Gardner’s charge to it’s logical conclusion and consider switches that behave like arbitrary “objects that have more than two positions”.

Another generalization of this puzzle could look at other ways of “spinning” the switches. In 1995, Ehrenborg and Skinner [5] did this in a puzzle they call ”Blind Bartender with Boxing Gloves”, that analyzed this puzzle while allowing the adversary to use an arbitrary, faithful group action to “scramble” the switches. We analyze our generalized switches within this same context.

This puzzle was re-popularized in 2019 when it appeared in “The Riddler” column from the publication FiveThirtyEight [6]. Shortly after this, in 2022, Yuri Rabinovich synthesized Bar Yehuda and Ehrenborg’s results in a a paper that modeled the collection of switches as a vector space over a finite field, and modeled the “spinning” or “scrambling” as a faithful, linear group action.

Sidana [7] provides a detailed overview of the history of this and related problems.

1.1.2 A Solution to the Winkler’s Spinning Switches Puzzle

We will start by discuss the solution to Winkler’s version of the puzzle because the solution provides some insights and intuition for the techniques that we use later. Before solving the four-switch version of the puzzle, we will make Peter Winkler proud by beginning with a simpler, two-switch version.

Example 1.1.1. *Suppose that we have two identical unlabeled switches on opposite corners of a square table, as in Figure 1.1*

Then we have a three-step solution for solving the problem. We start by toggling both switches simultaneously. If this does not turn on the light, this means that the switches were (and still) are in different states.

Then, the adversary spins the table. Next, we toggle one of the two switches to ensure that the switches are both in the same state. If the light hasn’t turned on, both must be in the off state.

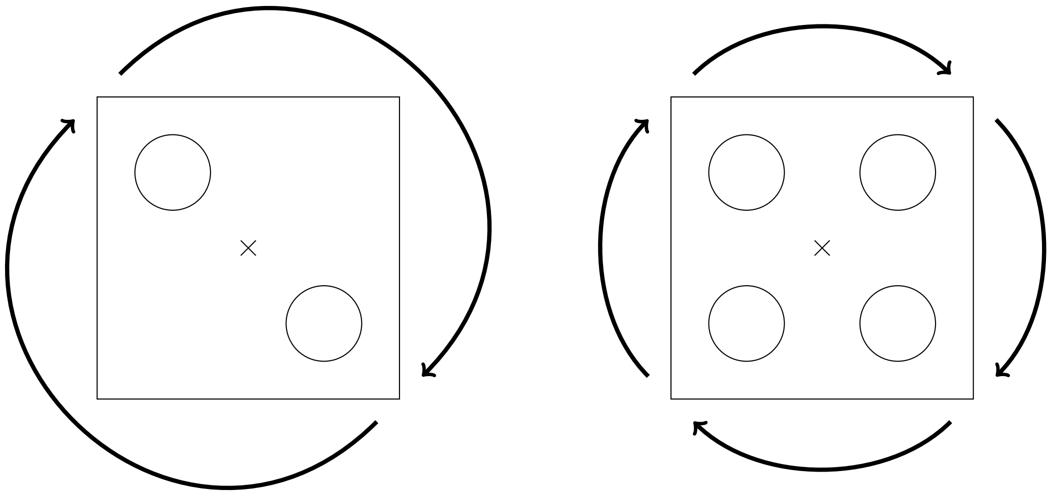

Figure 1.1: Illustration of both the two-switch and Winkler's original four-switch version of the puzzle, both on a spinning square table.

The adversary spins the table once more, but to no avail. We know both switches are in the off state, so we toggle them both simultaneously, turning on the lightbulb.

In order to bootstrap the two-switch solution into a four-switch solution, we must notice two things:

1. First, if we can get two switches along each diagonal into the same state respectively, then we can solve the puzzle by toggling both diagonals (all four switches), both switches in a single diagonal, and both diagonals again. In this (sub-)strategy, toggling both switches along a diagonal is equivalent to toggling a single single switch in the above example.

2. Second, we can indeed get both diagonals into the same state by toggling a switch from each diagonal (two switches on any side of square), then a single switch from one diagonal, followed by a switch from each switch.

We will interleave these strategies in a particular way, following the notation of Rabinovich [8].

Definition 1.1.2. *Given two sequences $A = \{a_i\}_{i=1}^N$ and $B = \{b_i\}_{i=1}^M$, we can define the* ***interleave*** *operation as*

$$A \circledast B = (A, b_1, A, b_2, A, \dots, b_M, A) \tag{1.1}$$

$$= (\underbrace{a_1, a_2, \dots, a_N}_{A}, b_1, \underbrace{a_1, a_2, \dots, a_N}_{A}, b_2, \underbrace{a_1, a_2, \dots, a_N}_{A}, \dots, b_M, \underbrace{a_1, a_2, \dots, a_N}_{A}). \tag{1.2}$$

which has length $(M+1)N + M = MN + M + N$.

Typically it is useful to interleave two strategies when A solves the puzzle given that the switches are in a particular state, and B gets the switches into that particular state. Usually, we also need A not to "interrupt" what B is doing. In the problem of four switches on a square table, B will ensure that the switches are in the same state within each diagonal, and A will turn on the light when that's the case. Moreover, A does not change the state within each diagonal.

Proposition 1.1.3. *There exists a fifteen-move strategy that guarantees that the light in Winkler's puzzle turns on.*

Proof. We begin by formalizing the two strategies. We will say that the first strategy S_1 where we toggle the two switches in a diagonal together will consist of the following three moves:

1. Switch **a**ll of the bulbs (A).

2. Switch the **d**iagonal consisting of the upper-left and lower-right bulbs (D).

3. Switch **a**ll of the bulbs (A).

We will say that the second strategy S_2 where we get the two switches within each diagonal into the same state consists of the following three moves:

1. Switch both switches on the left **s**ide (S).

2. Switch **one** switch (1).

3. Switch both switches on the left **s**ide (S).

Then the 15 move strategy is

$$S_1 \circledast S_2 = (A,S,A,D,A,S,A,1,A,S,A,D,A,S,A)$$

$\square$

We will generalize this construct in Theorem 1.4.1, which offers a formal proof that this strategy works.

It is worth briefly noting that $S_1 \circledast S_2$ is the fourth *Zimin word* (also called a *sequipower*), an idea that comes up in the study of combinatorics on words.

1.1.3 Generalizing Switches

Two kinds of switches are considered by Bar Yehuda, Etzion, and Moran in 1993 [4]: switches with a single "on" position that behave like n-state roulettes ($\mathbb{Z}_n$) and switches that behave like the finite field $\mathbb{F}_q$, both on a rotating k-gonal table. Yuri Rabinovich [8] goes further by considering collections of switches that behave like arbitrary finite dimensional vector spaces over finite fields that are acted on by a linear, faithful group action. We generalize this notion further by considering switches that behave like arbitrary finite groups.

Example 1.1.4. *In Figure 1.2, we provide a schematic for a switch that behaves like the symmetric group S_3. It consists of three identical-looking parts that need to be arranged in a particular order in order for the switch to be on.*

We could also construct a switch that behaves like the dihedral group of the square, D_8. This switch a flat, square prism that can slot into a square hole, and only one of the $|D_8| = 8$ rotations of the prism completes the circuit.

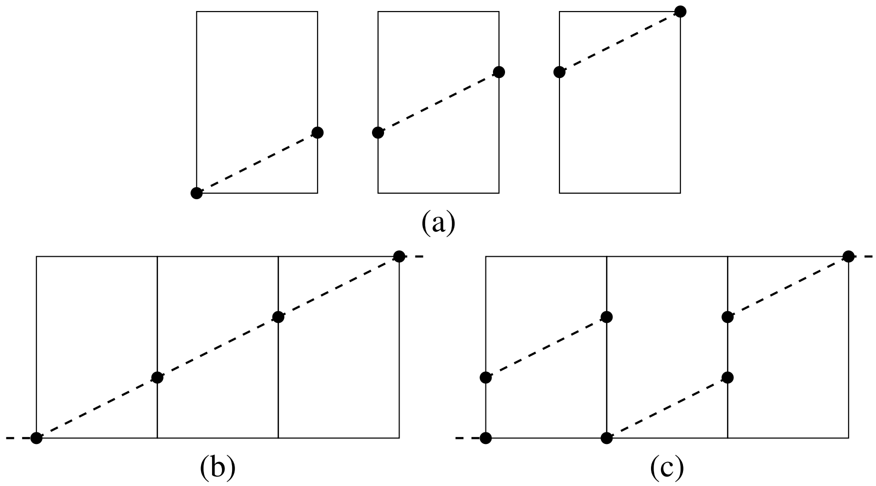

Figure 1.2: Part (a) shows a simple schematic for a switch that behaves like S_3, the symmetric group on three letters. The three rectangles can be permuted arbitrarily, but only configuration (b) completes the circuit. All other configurations fail to complete the circuit (e.g. (c)).

One subtlety of using a group G to model a switch is that both the "internal state" of a switch itself and the set of "moves" or changes are modeled by G. Perhaps we think of the state as the underlying set of G and the moves act via right group action of G on itself.

The reason that using a group to model a switch is because groups have many of the properties we would expect in a desirable switch.

Note 1.1.5. *The axioms for a group $(G, \cdot)$ closely follow what we would expect from a switch.*

1. (Closure) The group $(G, \cdot)$ is equipped with a binary operation, $\cdot\colon G \times G \to G$. That is, for all pairs of elements $g_1, g_2 \in G$ their product is in G

$$g_1 \cdot g_2 \in G.$$

In the context of switches, this means that if the switch is in some state $g_1 \in G$ and player B moves it with action $g_2 \in G$, then $g_1 \cdot g_2 \in G$ is a valid switch for the state.

2. (Identity) There exists an element $\mathrm{id}_G \in G$ such that for all $g \in G$,

$$\mathrm{id}_G \cdot g = g \cdot \mathrm{id}_G = g.$$

This axiom is useful because it means that Player B can “do nothing” to a switch and leave it in whatever state it is in. Because the identity is a distinguished element in G, we will also use the convention that id_G is the “on” or “winning” state for a given switch. (It is worth noting that all of the arguments work with small modification regardless of which element is designated as the on state.)

3. (Inverses) For each element $g \in G$ there exists an inverse element $g^{-1} \in G$ such that

$$g \cdot g^{-1} = g^{-1} \cdot g = \mathrm{id}_G .$$

This axiom states that no matter what state a switch is in, there is a move that will transition it into the on state.

4. (Associativity) Given three elements $g_1, g_2, g_3 \in G$,

$$(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$$

This is axiom is not strictly necessary for modeling switches, but as we will see in a later definition, it gives us a convenient way to describe the conditions for a winning strategy. (In Subsection 1.6.3, we briefly discuss dropping the associativity axiom by considering switches that behave like quasigroups with identity.)

1.1.4 Generalizing Spinning

We can also consider generalizations of “spinning” the switches. In particular, we will adopt the generalization from Ehrenborg and Skinner’s [5] 1995 paper, which use arbitrary faithful group actions to permute the switches. In particular, they provide a criterion that determines which group actions yield a winning strategy in the case of a given number of “ordinary” switches (those that behave like $\mathbb{Z}_2$). Rabinovich [8] stretches these results a bit further and looks at faithful linear group actions on collections of switches that behave are modeled as a finite dimensional vector space over a finite field. We build on this result in the context of more general switches.

1.2 The Wreath Product Model

Peter Winkler's version of the puzzle consists of four two-way switches on the corners of a rotating square table. The behavior of the switches are naturally modeled as $\mathbb{Z}_2$, and the rotating table is modeled as the cyclic group C_4. We will take the wreath product of $\mathbb{Z}_2$ by C_4 in order to get a mathematical model of the generalized spinning switches puzzle.

1.2.1 Modeling Generalized Spinning Switches Puzzles

We don't evoke wreath products arbitrarily: we use them because they are the right abstraction to model a generalized spinning switches puzzle where G describes the behavior of the switches, Ω describes the positions of the switches, and the action of H on Ω models the ways the adversary can permute the switches.

Definition 1.2.1 ([9])**.** *Let G and H be groups, let Ω be a finite H-set, and let $K = \prod_{\omega \in \Omega} G_\omega$, where $G_\omega \cong G$ for all $\omega \in \Omega$. Then the* **wreath product** *of G by H denoted by $G \wr H$, is the semidirect product of K by H, where H acts on K by $h \cdot (d_\omega) = d_{h^{-1}\omega}$ for $g \in H$ and $(g_\omega) \in \prod_{\omega \in \Omega} G_\omega$. The normal subgroup K of $G \wr H$ is called the* **base** *of the wreath product.*

The group operation is $(k,h) \cdot (k',h') = (k(h \cdot k'), hh')$

An element of $(k,h) \in G \wr H$ represents a turn of the game: The puzzle-solver chooses an element of the base $k \in K$ to indicate how they want to modify each of their switches and then their adversary chooses $h \in H$ and acts with h on Ω to permute the switches.

Example 1.2.2. *Consider the setup in the Winkler's version of the puzzle that consists of two-way switches ($\mathbb{Z}_2$) on the corners of a rotating square ($C_4 \cong \langle 0°, 90°, 180°, 270° \rangle$). The game itself corresponds to the wreath product $\mathbb{Z}_2 \wr C_4$. We will use the convention that the base of the wreath*

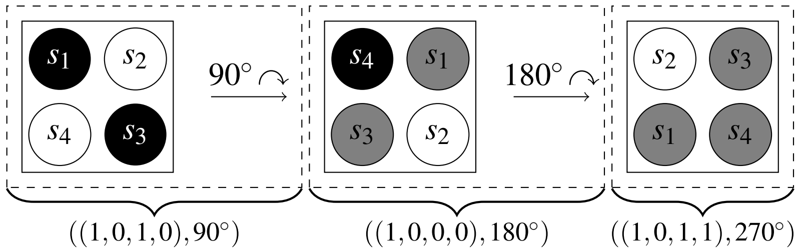

Figure 1.3: An illustration of two turns each in the Spinning Switches puzzle, modeled as elements of a wreath product.

product, K is ordered upper-left, upper-right, lower-right, lower-left; the group action is specified by degrees in the clockwise direction.

Consider the following two turns:

1. *During the first turn, the puzzle-solver toggles the upper-left and lower-right switches, and the adversary rotates the table $90°$ clockwise. This is represented by the element*

$$((1,0,1,0),90°) \in \mathbb{Z}_2 \wr C_4.$$

2. *During the second turn, the puzzle-solver toggles the upper-left switch, and the adversary rotates the table $90°$ clockwise. This is represented by the element*

$$((1,0,0,0),180°) \in \mathbb{Z}_2 \wr C_4.$$

As illustrated in Figure 1.3, the net result of these two turns is the same as a single turn where the puzzle-solver toggles the upper-left, upper-right, and lower-left switches and the adversary rotates the board $270°$ clockwise.

The multiplication under the wreath product agrees with this:

$$\begin{aligned} ((1,0,1,0),90°) \cdot ((1,0,0,0),180°) &= ((1,0,1,0) + \underbrace{90° \cdot (1,0,0,0)}_{(0,0,0,1)}, 90° + 180°) \\ &= ((1,0,1,1),270°) \end{aligned}$$

Occasionally it is useful to designate a particular state of the switches as the "on" state or the winning state. We will use the convention that the lightbulb turns on when all of the switches are equal to the identity, that is $\mathrm{id}_K \in K$. It is worth noting, however, that 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.

1.2.2 Switching Strategy

We will begin by formalizing the notation of a winning strategy in a generalized spinning switches puzzle. Informally, a switching strategy is a sequence of moves that the puzzle-solver can make that will put the switches into every possible state, which ensures the the "on" state is reached regardless of the initial (hidden) state of the switches.

Definition 1.2.3. *A* ***switching strategy*** *for $G \wr H$ is a finite sequence of elements in the base K, $\{k_i \in K\}_{i=1}^N$, such that for every sequence of elements in H, $\{h_i \in H\}_{i=1}^N$,*

$$p(\{\underbrace{e_{G\wr H}}_{m_0}, \underbrace{(k_1,h_1)}_{m_1}, \underbrace{(k_1,h_1)\cdot(k_2,h_2)}_{m_2}, \cdots, \underbrace{(k_1,h_1)\cdot(k_2,h_2)\cdots(k_N,h_N)}_{m_N}\}) = K.$$

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

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. (This follows the convention that $\mathrm{id}_K \in K$ is designated as the "on" state. If k' is chosen to the "on" state, then the sequence must contain $k^{-1}k'$.)

Proposition 1.2.4. *A finite sequence of moves is guaranteed to reach the "on" state if and only if it is a switching strategy.*

Proof. Without loss of generality, say that the "on" state for the switches is id_K. In the puzzle, we have an initial (hidden) state, k. Thus, after the i-th move, the wreath product element that represents the state of the switches is

$$p\left((k,\mathrm{id}_H)\cdot(k_1,h_1)\cdot(k_2,h_2)\cdots(k_i,h_i)\right) = k\cdot p\left((k_1,h_1)\cdot(k_2,h_2)\cdots(k_i,h_i)\right),$$

by associativity. We can factor out the first term because the "spin" is id_H, which acts trivially: $(k,\mathrm{id}_H)\cdot(k',h') = (kk',h')$

The initial state can be any $k \in K\setminus\{\mathrm{id}_K\}$, and this isn't known to the puzzle-solver. In order to reach the "on" state, there must exist some i, such that $p\left((k_1,h_1)\cdot(k_2,h_2)\cdots(k_i,h_i)\right) = k^{-1}$. k and adversarial sequences $\{h_i\}_{i=1}^{N}$. □

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 or randomly.

1.2.3 Bounds on the length of switching strategies

One useful consequence of this definition is that it quite straightforward to prove certain propositions. For example, the minimum length for a switching strategy has a simple lower bound.

Proposition 1.2.5. *Every switching strategy $\{k_i \in K\}_{i=1}^{N}$ is a sequence of length at least $|K|-1$.*

Proof. This follows from an application of the Pigeonhole Principle. Because the set

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

has at most $N+1$ elements. In order for the projection to be equal to K,

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

it must be the case that $N+1 \geq |K|$. Therefore $N \geq |K|-1$. $\square$

Minimal length switching strategies are common, so we give them a name.

Definition 1.2.6. *A **minimal switching strategy** for $G \wr H$ is a switching strategy of length $N = |K|-1$.*

In practice, every wreath product known by the author to have a switching strategy also has a known minimal switching strategy. In Section 1.6, we ask whether this property always holds.

1.3 Reductions

In this section, we develop examples of generalized spinning switches puzzles that do not have switching strategies using three techniques: directly, by a reduction on switches, or by a reduction on spinning.

1.3.1 Puzzles Known to Have No Switching Strategies

Our richest collection of known puzzles without switching strategies comes from a theorem of Rabinovich, which models switches as a vector space over a finite field.

Theorem 1.3.1. *[8] Assume that a finite "spinning" group H acts linearly and faithfully on a collection of switches that behave like a vector space V over a finite field $\mathbb{F}_q$ of characteristic p. Then the resulting puzzle has a switching strategy if and only if H is a p-group.*

It's worth noting that Rabinovich's switches are less general than arbitrary finite groups, but the "spinning" is more general: in addition to permuting the switches, the group action might add linear combinations of them as well.

Example 1.3.2. *By the theorem of Rabinovich [8], the game $\mathbb{Z}_2 \wr C_3$ does not have a switching strategy. In Rabinovich's notation, the vector space of switches is $\mathbb{Z}_2^3$ over the field $\mathbb{F}_2 = \mathbb{Z}_2$.*

The wreath product $\mathbb{Z}_2 \wr C_3$ is perhaps the simplest example of a generalized spinning switches puzzle without a switching strategy, so we will continue to use it as a basis of future examples.

1.3.2 Reductions on Switches

With Theorem 1.3.1 providing a family of wreath products without spinning switches to reduce to, we now introduce a theorem that allows us to prove that large families of wreath products also do not have a switching strategy.

Theorem 1.3.3. *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. We will prove the contrapositive, and suppose that $G' \wr H$ has base K' and a switching strategy $\{k'_i \in K'\}_{i=1}^{N}$.

The quotient map $\varphi\colon G' \mapsto G$ extends coordinatewise to $\varphi\colon K' \mapsto K$, which further extends in the first coordinate to $G \wr H$: $\varphi(k,h) := (\varphi(k),h)$.

It is necessary to verify that $\varphi\colon G' \wr H \to G \wr H$ is indeed a homomorphism.

$$\begin{aligned}
\varphi((k'_\alpha,h_\alpha)) \cdot \varphi((k'_\beta,h_\beta)) &= (\varphi(k'_\alpha),h_\alpha) \cdot (\varphi(k'_\beta),h_\beta) \\
&= (\varphi(k'_\alpha)(h_\alpha \cdot \varphi(k'_\beta)), h_\alpha h_\beta) \\
&= (\varphi(k'_\alpha)\varphi(h_\alpha \cdot k'_\beta), h_\alpha h_\beta) \\
&= (\varphi(k'_\alpha(h_\alpha \cdot k'_\beta)), h_\alpha h_\beta) \\
&= \varphi((k'_\alpha(h_\alpha \cdot k'_\beta), h_\alpha h_\beta)) \\
&= \varphi((k'_\alpha,h_\alpha) \cdot (k'_\beta,h_\beta))
\end{aligned}$$

Therefore the sequence $\{\varphi(k'_i) \in K\}_{i=1}^{N}$ is a switching strategy on $G \wr H$, because the quotient map $\varphi\colon G' \to G$ (and thus $\varphi\colon K' \to K$) is injective. $\square$

Example 1.3.4. *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, as illustrated in Figure 1.4.*

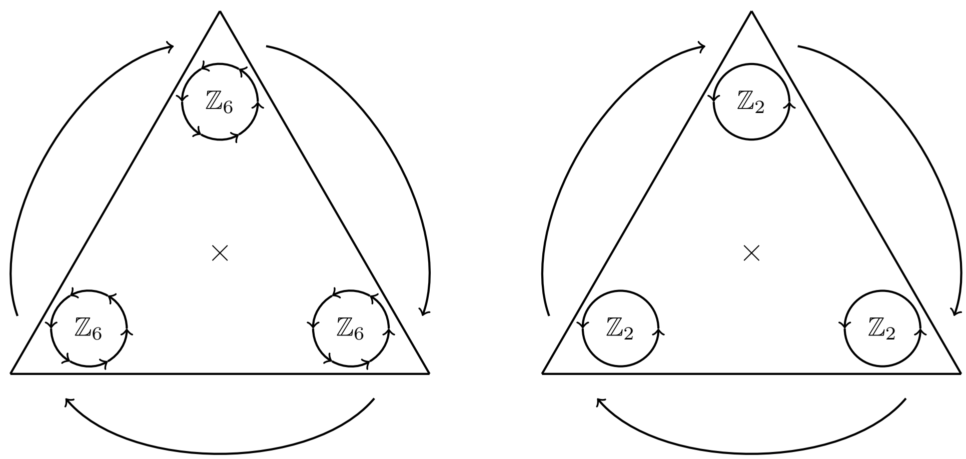

Figure 1.4: A reduction on switches: $\mathbb{Z}_6 \wr C_3$ reduces to $\mathbb{Z}_2 \wr C_3$, which is known not to have a switching strategy.

1.3.3 Reductions on Spinning

We can do two similar reductions on the "spinning" group of a wreath product. These theorems say that if a given wreath product $G \wr H$ does not have a switching strategy, then a similar wreath product $G \wr H'$ with a "more complicated" spinning group H' will not have a switching strategy either.

Theorem 1.3.5. *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.*

Proof. Again we will 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 $G \wr H$. $\square$

Example 1.3.6. *Consider the wreath product $\mathbb{Z}_2 \wr_{\Omega_6} C_3$ where Ω' consists of six switches on the corners of a hexagon as illustrated in Figure 1.3.6. While the group action of C_3 on Ω' is not*

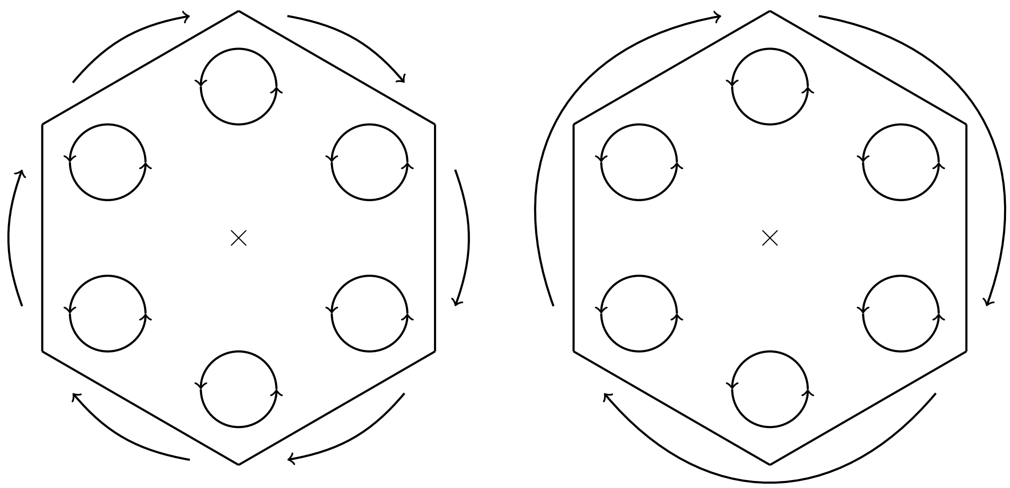

Figure 1.5: If there were a solution to $\mathbb{Z}_2 \wr_{\Omega_6} C_6$, then there would be a solution to $\mathbb{Z}_2 \wr_{\Omega_6} C_3$.

transitive, we know that $\mathbb{Z}_2 \wr_{\Omega_6} C_3$ *does not have a switching strategy, because in particular there is no way to ensure that the top, bottom-right, and bottom-left switches hit every state.*

Since $\mathbb{Z}_2 \wr_{\Omega_6} C_3$ *doesn't have a switching strategy,* $\mathbb{Z}_2 \wr_{\Omega_6} C_6$ *cannot not have a switching strategy either.*

In the above example, we noted that $\mathbb{Z}_2 \wr_{\Omega_6} C_3$ does not have a switching strategy by focusing on a triangle of switches and using the knowledge that $\mathbb{Z}_2 \wr C_3$ (two-way switches on a rotating triangular board) does not have a switching strategy. The following theorem allows us to take that very shortcut.

Theorem 1.3.7. *Suppose that* H' *is a group with a subgroup* $A \leq H'$ *such that* $A \cong H$*, and let*

$$\mathrm{Orb}(\omega) = \{\omega \cdot a : a \in A\} \subseteq \Omega$$

be the (right) orbit of $\omega \in \Omega$ *under* A*. If* $G \wr_{\mathrm{Orb}(\omega)} H$ *does not have a switching strategy, then* $G \wr_{\Omega} H'$ *does not have a switching strategy.*

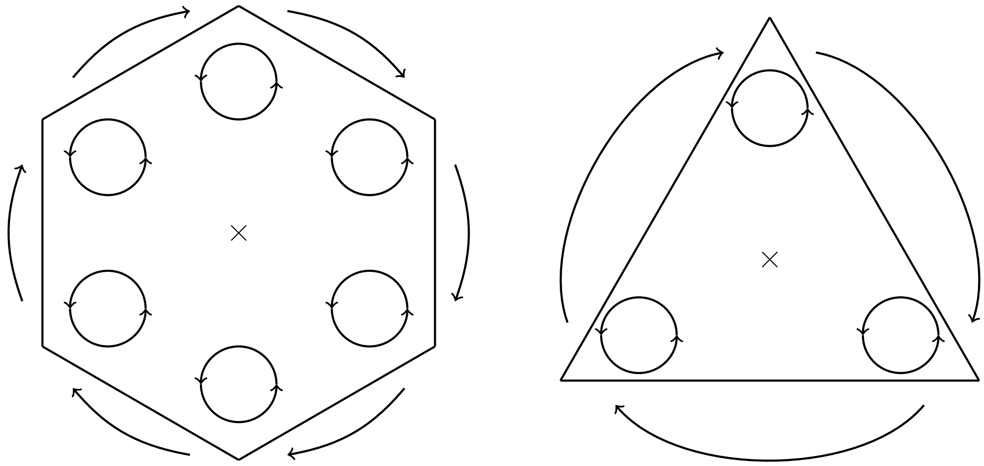

Figure 1.6: We know that $\mathbb{Z}_2 \wr_\Omega C_6$ cannot have a switching strategy, because that would imply a switching strategy for $\mathbb{Z}_2 \wr_{\Omega'} C_3$, where Ω' is the orbit of the top switch rotations of multiples of $120°$.

Proof. We start by making the contrapositive assumption that $G \wr_\Omega H'$ has a switching strategy $\{k_i \in K\}_{i=1}^N$, and we consider the projection $p_\omega \colon K \to K_\omega$ where

$$K = \prod_{\omega' \in \Omega} G_{\omega'} \qquad \text{and} \qquad K_\omega = \prod_{\omega' \in \mathrm{Orb}(\omega)} G_{\omega'}.$$

Then $\{p_\omega(k_i) \in K_\omega\}_{i=1}^N$ is a switching strategy for $G \wr_{\mathrm{Orb}(\omega)} H$, since the projection is a surjective map. $\square$

Example 1.3.8. *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, as illustrated in Figure 1.6.*

Now that we've proven that large families of wreath products do not have switching strategies, it's worthwhile to construct families of wreath products that do have switching strategies.

1.4 Switching Strategies on p-Groups

In this section, we'll develop a broad family of switching strategies, namely those where G and H (and thus $G \wr H$) are p-groups.

1.4.1 Switching Strategy Decomposition

Our first constructive theorem provides a technique that can be used to construct switching strategies for switches that behave like a group G in terms of a normal group and its corresponding quotient group.

Theorem 1.4.1. *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 $N \wr H$.

We ultimately would like to interleave these two strategies, but $k_i^{G/N} \notin K_G$. To find the appropriate analog, we partition G into $[G : N] = m$ right cosets of N,

$$G = Ng_1 \sqcup Ng_2 \sqcup \cdots \sqcup Ng_m,$$

each with a chosen representative in G. Now we define a map $r\colon G/N \to G$ that chooses the chosen representative of the coset, and extends coordinatewise. We use this map to define a sequence $S = \{r(k_i^{G/N}) \in K_G\}$.

We claim that these two sequences interleaved, $S_N \circledast S$, is a switching strategy for $G \wr H$. To prove this claim, we observe two facts:

1. Multiplying by elements of S_N will not change cosets and will walk through every element of its coset.

2. Multiplying by elements of S will walk through all cosets.

Therefore the interleaved sequence will walk through all elements of each coset, and thus is surjective onto K. $\square$

1.4.2 Construction of switching strategies on p groups

We start with a corollary of Theorem 1.3.1.

Corollary 1.4.2. *If H is a finite p-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 $n = 1$, $G \cong \mathbb{Z}_p \cong \mathbb{F}_p$, then there exists a switching strategy by Theorem 1.3.1. This is because H permutes the coordinates of $V = \mathbb{F}_p^{|\Omega|}$, and so it is a linear action on the vector space.

Otherwise, G is not simple. This means that G has a normal subgroup N of order $|N| = p^t$ (with $t \geq 1$) and a quotient G/N with order $|G/N| = p^{n-t}$. Because $t < n$ and $n - t < n$, we eventually end up at the $n = 1$ case by induction. $\square$

This means that whenever G and H (and thus $G \wr H$) are p-groups, then $G \wr H$ has a switching strategy.

1.4.3 A folklore conjecture

Here we note a conjecture from folklore, which—if true—implies that we have *almost* solved the problem in its full generality.

Conjecture 1.4.3 (Folklore)**.** *Almost all groups are* 2*-groups.*

One reason for this conjecture is computational. According to the On-Line Encyclopedia of Integer Sequences [10], there are $A000001(2^{10}) = 49487367289$ groups of order 2^{10} and there are $A063756(2^{11} - 1) = 49910536613$ groups of order less than 2^{11}. This means that more than 99.15% of the groups of order less than 2^{11} are of order 2^{10}.

If this conjecture is true, then most types of switches have switching strategies on most kinds of faithful finite group actions. Of course, while most finite groups may be 2-groups, most mathematicians are more interested in groups that *aren't*. This next section develops two families of examples of switching strategies where the switches do not behave like p-groups.

1.5 Switching strategies on other wreath products

Other authors have given switching strategies for various configurations of generalized spinning switches strategies, but in all of these examples, the wreath products themselves p-groups: $|G \wr_\Omega H| = |G|^{|\Omega|} \cdot |H|$, where H acts faithfully. In this section we introduce two families of wreath products that have switching strategies, but are not p-groups.

1.5.1 The trivial wreath product, $G \wr \mathbf{1}$

The simplest—and least interesting—way to construct a wreath product with a switching strategy is to remove the adversary (or randomness) altogether, by letting the spinning group be the trivial group $H = \mathbf{1}$. Additionally we will consider the case where $|\Omega| = 1$, so there is only one switch. Because the adversary cannot "spin" the switches at all, the puzzle-solver has perfect information the entire time. We will see that whenever G is a finite group, $G \wr \mathbf{1} \cong G$ has not just one switching strategy, but many.

Proposition 1.5.1. *The wreath product $G \wr \mathbf{1}$ has $(|G|-1)!$ minimal switching strategies.*

Proof. There are $(|G|-1)!$ permutations of $G \setminus \{\mathrm{id}_G\}$, and we claim each one corresponds to a minimal switching strategy. Namely, if $(k_1, k_2, \ldots, k_{|G|-1})$ is such a permutation, then the sequence $\{k'_i\}_{i=1}^{|G|-1}$ where $k'_1 = k_1$ and $k'_i = k_{i-1}^{-1} k_i$ is a switching strategy on $G \wr \mathbf{1}$.

Then we claim by induction that $(k'_1, \mathrm{id}) \cdot (k'_2, \mathrm{id}) \cdots (k'_j, \mathrm{id}) = (k_j, \mathrm{id})$. By construction, the base case is true when $j = 1$. If the claim holds up to $j-1$, then

$$\underbrace{(k'_1, \mathrm{id}) \cdot (k'_2, \mathrm{id}) \cdots (k'_{j-1}, \mathrm{id})}_{(k_{j-1}, \mathrm{id})}(k'_j, \mathrm{id}) = (k_{j-1}, \mathrm{id})(k_{j-1}^{-1} k_j, \mathrm{id}) = (k_j, \mathrm{id}),$$

as desired. Thus the projection of the partial products is

$$p(\{\underbrace{e_{G\wr\mathbf{1}}}_{m_0},\underbrace{(k_1',h_1)}_{m_1},\underbrace{(k_1',h_1)\cdot(k_2',h_2)}_{m_2},\cdots,\underbrace{(k_1',h_1)\cdot(k_2',h_2)\cdots(k_N',h_N)}_{m_N}\}) \tag{1.3}$$

$$= p(\{e_{G\wr\mathbf{1}},(k_1,\mathrm{id}),(k_2,\mathrm{id}),\cdots,(k_N',\mathrm{id})\}) \tag{1.4}$$

$$= \{e_{G\wr\mathbf{1}},k_1,k_2,\cdots,k_{|G|-1}\} = K, \tag{1.5}$$

where $\{k_1,k_2,\ldots,k_{|G|-1}\}$ spans $G\setminus\{\mathrm{id}_G\}\cong K\setminus\{\mathrm{id}_K\}$ by assumption. □

While the trivial wreath product is a useful example to keep in mind for generating counterexamples, we're generally more interested in the situation where the adversary permutes the switches to create uncertainty for the puzzle-solver.

1.5.2 Two interchangeable groups generated by involutions

In this section, we will construct a switching strategy for the generalized spinning switches puzzle that consists of two switches that behave like a group G that can be generated by involutions, which the adversary can swap. (See, for example, Figure 1.2 which illustrates switches that behave like $S_3 = \langle(12),(13)\rangle$)

This strategy relies on the fact that because these generators are their own inverse, applying a generator to the first switch or the second switch has no effects on the difference.

This switching strategy has two parts. The first part ensures that the two switches have every possible difference. The second part show that we can get the first switch (with respect to the projection onto K) to take on every possible value without changing the difference.

Theorem 1.5.2. *Suppose that G is a group that can be generated by involutions. Then the generalized spinning switches puzzle consisting of two, interchangeable copies of G, namely $G\wr C_2$, has a switching strategy.*

Proof. We start by writing G in terms of its generators: $G = \langle t_1, t_2, \ldots, t_N \rangle$, where $t_i^{-1} = t_i$. Because this is the generating set, there exists a finite sequence of transpositions $(t_{i_1}, t_{i_2}, \ldots, t_{i_M})$ such that the partial products of the sequence generate G:

$$G = \{\mathrm{id}_G, t_{i_1}, t_{i_1}t_{i_2}, \ldots, t_{i_1}t_{i_2}\cdots t_{i_M}\}.$$

We develop the strategy in two parts. First, we provide a strategy $A = \{\alpha_i \in K\}$ such that for any adversarial sequence $\{h_i \in H\}$ and element $g \in G$, there exists an $i \geq 0$ such that the i-th partial product, $(g_{i,1}, g_{i,2}) = p((\mathrm{id}_K, \mathrm{id}_H) \cdot (\alpha_1, h_1) \cdot (\alpha_2, h_2) \ldots (\alpha_i, h_i))$, has a difference of g. That is, $g_{i,1}g_{i,2}^{-1} = g$.

To do this, we define $\alpha_j = (t_{i_j}, \mathrm{id}_G)$, and notice that the difference of the coordinates is the same whether we add α_j or $(180°) \cdot \alpha_j$ to a element $(g_1, g_2) \in K$:

$$g_1(g_2 t_{i_j})^{-1} = g_1 t_{i_j}^{-1} g_2^{-1} = (g_1 t_{i_j}) g_2^{-1}$$

Because the partial products of t_{i_j} cover G, there exists some $t_1 t_2 \ldots t_k = g_1 {-} 1 g g^2$, so that $g_{i,1}g_{i,2}^{-1} = g$, as desired.

Next, we give a strategy $B = \{\beta_i\}$ such that for any adversarial sequence $\{h_i \in H\}$ and element $g \in G$, there exists an $i \geq 0$ such that the i-th partial product, $(g_{i,1}, g_{i,2}) = p((\mathrm{id}_K, \mathrm{id}_H) \cdot (\alpha_1, h_1) \cdot (\alpha_2, h_2) \ldots (\alpha_i, h_i))$ has a first coordinate $g_{i,1} = g$.

To do this, we define $\beta_j = (t_{i_j}, t_{i_j})$. This strategy is invariant up to actions of H, so we can see that regardless of the initial state $(g_1, g_2) \in K$, there exists some j such that $\beta_1 \beta_2 \cdots \beta_j = g_1^{-1} g$ and therefore $(g_1, g_2)\beta_1\beta_2 \cdots \beta_j = (g, g_2 g_1^{-1} g)$, as desired.

It is important to note that applying β_j does not affect the difference: $(g_1, g_2)\beta_j = (g_1 t_{i_j}, g_2 t_{i_j})$ has a difference of $(g_1 t_{i_j})(g_2 t_{i_j})^{-1} = g_1 t_{i_j} t_{i_j}^{-1} g_2^{-1} = g_1 g_2^{-1}$.

Now by interleaving these two strategies, we see that the partial products of $B \circledast A$ hit every possible first letter and every possible difference for every first letter, therefore the projection of the partial products of $B \circledast A$ cover K for all adversarial sequences, so $B \circledast A$ is a switching strategy. $\square$

We will illustrate this idea explicitly letting G be the smallest nonabelian group of composite order, the symmetric group on three letters S_3, which is isomorphic to the dihedral group of the triangle, D_6.

Example 1.5.3. *Note that $S_3 = \langle (12), (13) \rangle$ is generated by involutions, and that the partial products of the sequence $((12), (13), (12), (13), (12))$ cover S_3.*

As a bit of notation, for a permutation $\pi \in S_3$, let $\pi_1 = (\pi, \mathrm{id}_{S_3}) \in K$ and $\pi_2 = (\pi, \pi) \in K$, corresponding to sequences B and A respectively in the above theorem. Then the following is a (minimal) switching strategy on $S_3 \wr C_2$:

$$
\begin{array}{ll}
(12)_2(13)_2(12)_2(13)_2(12)_2 & \\
 & (12)_1 \\
(12)_2(13)_2(12)_2(13)_2(12)_2 & \\
 & (13)_1 \\
(12)_2(13)_2(12)_2(13)_2(12)_2 & \\
 & (12)_1 \\
(12)_2(13)_2(12)_2(13)_2(12)_2 & \\
 & (13)_1 \\
(12)_2(13)_2(12)_2(13)_2(12)_2 & \\
 & (12)_1 \\
(12)_2(13)_2(12)_2(13)_2(12)_2 &
\end{array}
$$

It is natural to ask which generalized spinning switches puzzles with switches that symmetric groups have switching strategies. We can use a reduction on switches to rule out a large amount of these.

Proposition 1.5.4. *For $n > 1$, $S_n \wr H$ does not have a switching strategy whenever $|H|$ is not a power of 2.*

Proof. The alternating group A_n is an index 2 subgroup of S_n, so A_n is normal, and $S_n/A_n \cong \mathbb{Z}_2$. Since we know that $\mathbb{Z}_2 \wr H$ has no switching strategy when $|H|$ is not a power of 2, by the reduction in Theorem 1.3.3, $S_n \wr H$ does not have a switching strategy. $\square$

Many groups are generated by transpositions, including 22 of the 26 sporadic simple groups and the alternating group A_5 and A_n for $n > 9$, as shown by Mazurov and Nuzhin respectively.

Theorem 1.5.5. *[11] Let G be one of the 26 sporadic simple groups. The group G cannot be generated by three involutions two of which commute if and only if G is isomorphic to M_{11}, M_{22}, M_{23}, or M_L^c.*

Theorem 1.5.6. *[12] The alternating group A_n is generated by three involutions the two of which commute, if and only if $n \geq 9$ or $n = 5$.*

Nuzhin provides other families of groups that are generated by three involutions, two of which commute, in subsequent papers. [13–15]

Thus, we have characterized finite wreath products with switching strategies in many cases: wreath products that are p-groups, trivial wreath products, and wreath products $G \wr C_2$ where G is generated by involutions. In the next section, we provide even more general constructions and ask more specific questions.

1.6 Open questions

In this section, we provide conjectures and suggest open questions about the structure of switching strategies when they exist, further generalizations of spinning switches puzzles, and finally introduce a notion of an infinite switching strategy for infinite wreath products.

The ultimate open question is a full classification of finite wreath products with switching strategies.

OpenQuestion 1.6.1. *What finite wreath products, $G \wr H$, have a switching strategy?*

1.6.1 Switches generated by elements of prime power order

In Theorem 1.5.2, we constructed a strategy for $G \wr C_2$, when G can be generated by elements of order 2. We conjecture that that there is a broader construction for the case where the adversary can act on the switches with a group of order 2^{ℓ}.

Conjecture 1.6.2. *When G is a finite group generated by involutions, and H is 2-group that acts faithfully on the set of switches, there exists a switching strategy for $G \wr H$.*

We can also ask about three switches that are generated by elements of order 3 on the corners of a triangular table. In particular, the alternating group is such a group, and we conjecture that it has a switching strategy.

Conjecture 1.6.3. *There exists a switching strategy for $A_n \wr C_3$.*

Putting these two conjectures together, we boldly predict a large family of wreath products with switching strategies.

Conjecture 1.6.4. *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.*

1.6.2 Palindromic switching strategies

In all known examples, when there exists a switching strategy S, we also know of a *palindromic* switching strategy $S' = \{k'_i \in K\}_{i=1}^{N}$ such that $k'_i = k'_{N-i+1}$ for all i.

Conjecture 1.6.5. *Whenever $G \wr H$ has a switching strategy, it also has a palindromic switching strategy.*

If this conjecture is false, we suspect a counterexample can be found in the case of the trivial wreath product, $G \wr 1 \equiv G$.

1.6.3 Quasigroup switches

In Section 1.1.3, we argued for modeling switches as finite groups because of some 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.

In the list, we also included the axiom of associativity for three reasons: switches in practice typically have associativity, groups are easier to model than quasigroups with identity, and it makes the definition of a "switching strategy" less fussy.

However, one could design a switch that does not have associativity because, and the puzzle would still be coherent. This is because the process of a generalized spinning switches puzzle is naturally "left associative" in the language programming language theory: we are always "stacking" our next move onto the right. As such, it is worth noting the slightly more general way of modeling switches: as quasigroups with identity, called *loops*.

In particular, we are interested in the smallest loop that is not a group [16], which we denote $\mathscr{L} = (\{1,a,b,c,d\},*)$, and describe via its multiplication table, a Latin square of order 5. (Notice that $(ab)d = a \neq d = a(bd)$.)

$*$	1	a	b	c	d
1	1	a	b	c	d
a	a	1	c	d	b
b	b	d	1	a	c
c	c	b	d	1	a
d	d	c	a	b	1

Conjecture 1.6.6. *There exists a nontrivial adversarial group H such that the generalized spinning switches puzzle with switches that look like $\mathscr{L}$ has a winning strategy for the puzzle-solver.*

1.6.4 Expected number of turns

In practice, a puzzle-solver can "win" by playing randomly, which will *eventually* turn on the lightbulb with probability 1, thanks to the finite number of configurations and the law of large numbers.

While the puzzle asks for a solution in finite time, it is natural to ask about the expected value of the number of turns given various sequences of moves. Notice that this is an interesting question even (perhaps especially) in the context of generalized spinning switches puzzles that do not have a switching strategy.

Winkler [17] notes in the solution "Spinning Switches":

> Although no fixed number of steps can guarantee turning the bulb on in the three-switch version [with two-way switches], 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. [17]

A basic model for computing the expected number of turns assumes that the initial hidden state $k \in K$ is not the winning state id_K, and that the adversaries "spins" are independent and identically distributed uniformly random elements $h_j \in H$.

Proposition 1.6.7. *If the puzzle-solver chooses $k_j \in K \setminus \{\mathrm{id}_K\}$ uniformly at random (that is, never choosing the "do nothing" move) then the distribution of the resulting state will be uniformly distributed among the $|K|-1$ different states, the probability of the resulting state being the winning state is*

$$\mathbb{P}(p((k_1,h_1)(k_2,h_2)\dots(k_j,h_j)) = k^{-1} \mid p((k_1,h_1)(k_2,h_2)\dots(k_{j-1},h_{j-1}) \neq k^{-1})) = \frac{1}{|K|-1},$$

and the expected number of moves is $|K|-1$.

Proof. Because the new states are in 1-to-1 correspondence with the elements of $K \setminus \{\mathrm{id}_K\}$, since $k_j \in \setminus\{\mathrm{id}_K\}$ is chosen uniformly at random, $p((k_1,h_1)(k_2,h_2)\dots(k_j,h_j))$ is uniformly distributed among all elements of K besides the projection of the first $j-1$ elements. The expected value is $|K|-1$ because the number of turns follows a geometric distribution with parameter $(|K|-1)^{-1}$. $\square$

When a generalized spinning switches puzzle has a *minimal* switching strategy, we can *guarantee* that we turn on the light within $|K|-1$ moves, so we certainly can solve in fewer than $|K|-1$ moves on average.

Proposition 1.6.8. *If the generalized spinning switches puzzle, $G \wr H$, has a minimal switching strategy, then the expected number of moves is $|K|/2$.*

Proof. If there's a switching strategy of length $|K|-1$, then for each adversarial strategy $\{h_i \in H\}_{i=1}^{|K|-1}$ the projection of the sequence of partial products of moves induces a permutation of $K \setminus \{\mathrm{id}_K\}$.

If the initial hidden state k is chosen uniformly at random, then k^{-1} is equally likely to occur at any position in this permutation, so the index of the winning state is uniform on $\{1, 2, \ldots, |K|-1\}$ and so the expected number of moves is

$$\frac{1+2+\cdots+|K|-1}{|K|-1} = |K|/2.$$

$\square$

We can always come up with a strategy that does better than the strategy in Proposition 1.6.7.

Proposition 1.6.9. *For every generalized spinning switches puzzle, $G \wr H$ such that $|K| > 2$, there always exists a (perhaps infinite) sequence whose expected number of moves is strictly less than $|K|-1$.*

Proof. When $|K| > 2$, we can always improve on the random strategy in Proposition 1.6.7, by avoiding the move of $(g, g, ..., g) \in K$ followed by $(g^{-1}, g^{-1}, ..., g^{-1}) \in K$, because the second move will put us into a previous state and so will turn on the lightbulb with probability 0. $\square$

We conjecture that this technique can be extended, and that the puzzle-solver can always do asymptotically better than randomly guessing.

Conjecture 1.6.10. *There exists a constant $\frac{1}{2} < c < 1$ such that for all (finite) wreath products $G \wr H$ with sufficiently large $|K|$, the expected number of moves is less than $c|K|$.*

1.6.5 Bounds on the length of shortest switching strategies

Based on all of the examples that we know of, we conjecture that we can find minimal switching strategies whenever we can find one at all.

Conjecture 1.6.11. *Whenever $G \wr H$ has a switching strategy, it also has a minimal switching strategy $\{k_i \in K\}_{i=1}^{|K|-1}$.*

On the other extreme, we have a weaker conjecture: whenever a wreath product has a switching strategy, we can provide an upper bound for its shortest switching strategy.

Conjecture 1.6.12. *Let K/H be the set of equivalence classes of K up to the action of H. Then if $G \wr H$ has a switching strategy, it always has a switching strategy of length $N < 2^{|K/H|-1}$.*

1.6.6 Counting switching strategies

The counting problem analog to the decision problem "does $G \wr H$ have a switching strategy" is obviously of interest to this combinatorialist. It is interesting to count both the number of switching strategies of length N, and the number of such switching strategies *up to the action of H*. We might also be interested in the number of palindromic switching strategies or any other modifier.

In the case of the trivial wreath product, $G \wr \mathbf{1}$, we saw in Proposition 1.5.1 that there are $(|G| - 1)!$ switching strategies. (And since the group action is trivial, there are also this many strategies up to group action.)

OpenQuestion 1.6.13. *Given a wreath product $G \wr H$ how many switching strategies of length N does it have? How many up to the action of H? How many are palindromic?*

Proposition 1.6.14. *The wreath product $S_3 \wr \mathbf{1}$ has* 12 *palindromic switching sequences:*

$$\begin{array}{rrrrr}
(1\,2), & (1\,3), & (1\,2), & (1\,3), & (1\,2). \\
(1\,2), & (2\,3), & (1\,2), & (2\,3), & (1\,2). \\
(1\,3), & (1\,2), & (1\,3), & (1\,2), & (1\,3). \\
(1\,3), & (2\,3), & (1\,3), & (2\,3), & (1\,3). \\
(1\,2\,3), & (1\,2\,3), & (1\,2), & (1\,2\,3), & (1\,2\,3). \\
(1\,2\,3), & (1\,2\,3), & (1\,3), & (1\,2\,3), & (1\,2\,3). \\
(1\,2\,3), & (1\,2\,3), & (2\,3), & (1\,2\,3), & (1\,2\,3). \\
(1\,3\,2), & (1\,3\,2), & (1\,2), & (1\,3\,2), & (1\,3\,2). \\
(1\,3\,2), & (1\,3\,2), & (1\,3), & (1\,3\,2), & (1\,3\,2). \\
(1\,3\,2), & (1\,3\,2), & (2\,3), & (1\,3\,2), & (1\,3\,2). \\
(2\,3), & (1\,2), & (2\,3), & (1\,2), & (2\,3). \\
(2\,3), & (1\,3), & (2\,3), & (1\,3), & (2\,3).
\end{array}$$

Proof. The search space is small here, so this was computed naively by brute force. □

1.6.7 Multiple moves between each turn

Another way that we could generalize a spinning switches puzzle to provide winning strategies for the puzzle-solver is by restricting the adversary's moves. For instance, we could modify the puzzle so that the adversary's spinning sequence in such a way that the adversary can only permute the switches every k turns. For every finite setup $G \wr H$, there exists $k \in \mathbb{N}$ such that the puzzle-solver can win. (For example, we can always take $k > |K|$ so that the puzzle-solver can do a walk of K.)

OpenQuestion 1.6.15. *How does one compute the minimum k such that the puzzle solver has a switching strategy of $G \wr H$ given that the adversary's sequence $\{h_i \in H\}$ is constrained so that $h_i = e_H$ whenever $i \not\cong 0 \bmod k$?*

1.6.8 Nonhomogeneous switches

Another generalization of the spinning switches puzzle is by allowing different sorts of switches. For instance, we could imagine a square board containing four buttons: two 2-way switches, a

3-way switch, and a 5-way switch. Can a puzzle like this be solved? It is important that all of these buttons have the same "shape", that is there is some group G with a surjective homomorphism onto each of them.

One way to formalize this generalization is as follows:

Definition 1.6.16. *A* ***nonhomogeneous generalized spinning switches puzzle*** *consists of a triple*

- *A wreath product $\mathbb{F}_k \wr_\Omega H$ of the free group on k generators by a "rotation" group with a base denoted $K = \prod_{\omega\in\Omega} F_{k,\omega}$,*
- *a product of finite groups denoted $\hat{G} = \prod_{\omega\in\Omega} G_\omega$ where each group is specified by a presentation with k generators and any number of relations: $G_\omega = \langle g_1^\omega, g_2^\omega, \dots, g_k^\omega \mid R_\omega\rangle$, and*
- *a corresponding sequence of evaluation maps $e_\omega \colon \mathbb{F}_{k,\omega} \to G_\omega$, that send generators in $F_{k,\omega}$ to the corresponding generators in G_ω, and can be induced coordinatewise to a map $e_\Omega \colon K \to \hat{G}$.*

When all of the G_ωs are isomorphic, this essentially simplifies to the original definition.

The analogous definition of a switching strategy becomes more complicated.

Definition 1.6.17. *Let $(\mathbb{F}_k \wr_\Omega H, \hat{G}, e_\Omega)$ be a nonhomogeneous generalized spinning switches puzzle.*

Then a ***nonhomogeneous switching strategy*** *is a sequence $\{k_i \in K\}_{i=1}^N$ such that for each adversarial sequence $\{h_i \in H\}_{i=1}^N$ the induced projection/evaluation map $e_\Omega \circ p \colon \mathbb{F}_k \wr_\Omega H \to \hat{G}$ on the partial products of $\{(k_i, h_i) \in \mathbb{F}_k\}$ covers $\hat{G}$.*

Proposition 1.6.18. *In the specific case that $\Omega = [n]$, $H \subseteq S_n$, and $\hat{G} = \prod_{\omega\in\Omega} G_\omega$ is a product of cyclic groups of pairwise coprime order. Then the nonhomogeneous generalized spinning switches puzzle has a switching strategy, namely $\{(1, 1, \dots, 1) \in K\}_{i=1}^{|K'|-1}$.*

Proof. By the fundamental theorem of abelian groups, $\hat{G}$ is cyclic and is generated by

$$\hat{G} = \langle (1_{C_{k_1}}, 1_{C_{k_2}}, \dots, 1_{C_{k_{|\Omega|}}})\rangle.$$

Thus, the sequence $\{(1,1,\dots,1) \in K\}_{i=1}^{|K'|-1}$ is a switching strategy because it is a fixed point under H, and its image is a generator of $\hat{G}$. $\square$

OpenQuestion 1.6.19. *Which nonhomogeneous generalized spinning switches puzzles have a nonhomogeneous switching strategy?*

1.6.9 Infinite Switching Strategies

When we first introduced the notion of a switching strategy in Definition 1.2.3, we defined it to be a finite sequence on finite wreath products. However, we can expand the definition to (countably) infinite wreath products by allowing for infinite sequences of moves in K. In particular, we can extend this definition to settings where switches have a countably infinite number of states, where there are a countably infinite number of switches, or both. To keep K countable in the latter cases, we use the restricted wreath product, where $K \cong \bigoplus_{\omega \in \Omega} G_\omega$ is defined to be a direct sum instead of a direct product.

Definition 1.6.20. *A **infinite switching strategy** on an infinite wreath product $G \wr H$ is a sequence $\{k_i \in K\}_{i=1}^{\infty}$ such that for all $k \in K$ and all infinite sequences $\{h_i \in H\}_{i=1}^{\infty}$, there exists some $N \geq 0$ such that the projection*

$$p((k_1,h_1)\cdot(k_2,h_2)\cdots(k_N,h_N)) = k^{-1}.$$

We claim, but do not prove that, $G \wr_\Omega C_2$ has a infinite switching strategy in the following three settings:

1. $|\Omega| = 2$ and $G \cong (\mathbb{N}_{\geq 0}, \wedge)$ where $\wedge$ is the bitwise XOR operator.

2. $\Omega \cong \mathbb{N}_{\geq 0}$ as a set, and $K = \bigoplus_{i=0}^{\infty} \mathbb{Z}_2^{(i)}$ where $C_2^{(i)} \cong \mathbb{Z}_2$.

3. $\Omega \cong \mathbb{N}_{\geq 0}$ as a set, and $K = \bigoplus_{i=0}^{\infty} (\mathbb{N}_{\geq 0}, \wedge)$ for all i.

Chapter 2

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.

2.1 Background

In 2010, Mark Conger [18] 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 2.1) 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 2.2 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 2.3 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 2.3 emphasizes the case of S_{kn}, Section 2.4 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 2.2 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 2.4 constructs such a bijection explicitly, and proves that it has the desired properties.

2.1.1 Motivating Examples

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

Definition 2.1.1. *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 2.1.2. *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 2.1 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

		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	$7/1$	$7/1$	$7/1$	$7/1$	$7/1$	$7/1$	$7/1$

Table 2.1: A table of the expected value of the first letter of $\pi \in S_n$ with exactly m 2-cycles, $\mathbb{E}[\pi(1) \,|\, \pi \in S_n \text{ has exactly } m \text{ 2-cycles}]$.

position $(i+2, j+1)$, and that these denominators increase with n. The sequence of denominators begins

$$1, 5, 29, 233, 2329, 27949, \ldots, \tag{2.1}$$

which agrees with the type B derangement numbers, sequence A000354 in the On-Line Encyclopedia of Integer Sequences (OEIS) [10]. In other words, the denominators in the table appear to be related to the symmetries of the hypercube that move every facet.

2.2 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.

2.2.1 Counting permutations based on cycles

Theorem 2.2.1 ([19], [20])**.** *The number of permutations in S_n with exactly m k-cycles is given by the following sum, via the principle inclusion/exclusion:*

$$C_k(n,m) = \frac{n!}{m!k^m} \sum_{i=0}^{\lfloor n/k \rfloor - m} \frac{(-1)^i}{i!\,k^i}. \tag{2.2}$$

Corollary 2.2.2. *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.*

$$C_k(n,m) - nC_k(n-1,m) = \begin{cases} 0 & k \nmid n \quad (2.3a) \\ \dfrac{n!(-1)^{\frac{n}{k}-m}}{(n/k)!\,k^{\frac{n}{k}}}\dbinom{n/k}{m} & k \mid n \quad (2.3b) \end{cases}$$

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

$$\frac{n!}{m!k^m} \sum_{i=0}^{\lfloor n/k \rfloor - m} \frac{(-1)^i}{i!\,k^i} - n\left(\frac{(n-1)!}{m!k^m} \sum_{i=0}^{\lfloor (n-1)/k \rfloor - m} \frac{(-1)^i}{i!\,k^i}\right) = 0. \tag{2.4}$$

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

$$\begin{aligned}
\frac{n!}{m!k^m} \sum_{i=0}^{n/k-m} \frac{(-1)^i}{i!\,k^i} - n\left(\frac{(n-1)!}{m!k^m} \sum_{i=0}^{n/k-1-m} \frac{(-1)^i}{i!\,k^i}\right) \\
&= \frac{n!}{m!k^m}\left(\frac{(-1)^{n/k-m}}{(n/k-m)!k^{n/k-m}}\right) \\
&= \frac{n!(-1)^{n/k-m}}{(n/k-m)!m!k^{n/k}} \\
&= \frac{n!(-1)^{\frac{n}{k}-m}}{(n/k)!\,k^{n/k}}\binom{n/k}{m}. && (2.5)
\end{aligned}$$

□

See Section 2.4 for a bijective proof of Equation 2.3a.

2.2.2 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 2.2.3. *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

$$\mathbb{E}[\pi(1) \,|\, \pi \in S_n \text{ has exactly } m \text{ } k\text{-cycles}] = \frac{1}{C_k(n,m)} \sum_{a=1}^{n} a C_k^{(a)}(n,m). \qquad (2.6)$$

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

Proposition 2.2.4. *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:*

$$C_k^{(1)}(n,m) = C_k(n-1,m). \tag{2.7}$$

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 $k > 1$. $\square$

Proposition 2.2.5. *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:*

$$C_k^{(2)}(n,m) = \cdots = C_k^{(a)}(n,m) = \cdots = C_k^{(b)}(n,m) = \cdots = C_k^{(n)}(n,m). \tag{2.8}$$

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. $\square$

Lemma 2.2.6. *For all $2 \leq a \leq n$,*

$$C_k^{(a)}(n,m) = \frac{C_k(n,m) - C_k(n-1,m)}{n-1}. \tag{2.9}$$

Proof. Since

$$C_k(n,m) = C_k^{(1)}(n,m) + C_k^{(2)}(n,m) + \cdots + C_k^{(n)}(n,m), \tag{2.10}$$

using Proposition 2.2.5, for the last $(n-1)$ terms, this can be rewritten as

$$C_k(n,m) = C_k^{(1)}(n,m) + (n-1)C_k^{(a)}(n,m). \tag{2.11}$$

Solving for $C_k^{(a)}(n,m)$ and using the substitution from Proposition 2.2.4 gives the desired result. $\square$

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.

2.2.3 Expected value of first letter

Theorem 2.2.7. *For $k > 1$, the expected value of the first letter of a permutation $\pi \in S_n$ with m k-cycles is given by*

$$\begin{aligned}
&\mathbb{E}[\pi(1) \mid \pi \in S_n \textit{ has exactly } m \textit{ k-cycles}] \\
&\qquad\qquad = \frac{n}{2}\left(1 - \frac{C_k(n-1,m)}{C_k(n,m)}\right) + 1. \qquad (2.12)
\end{aligned}$$

Proof. Using Proposition 2.2.5, we can consolidate all but the first term of the sum in Equation 2.6

$$\begin{aligned}
&\sum_{a=1}^{n} aC_k^{(a)}(n,m) && (2.13)\\
&= C_k^{(1)}(n,m) + \sum_{a=2}^{n} aC_k^{(n)}(n,m) && (2.14)\\
&= C_k^{(1)}(n,m) + \frac{(n-1)(n+2)}{2}C_k^{(n)}(n,m) && (2.15)\\
&= C_k(n-1,m) + \frac{(n-1)(n+2)}{2}\left(\frac{C_k(n,m) - C_k(n-1,m)}{n-1}\right) && (2.16)\\
&= \left(\frac{n}{2}+1\right)C_k(n,m) - \frac{n}{2}C_k(n-1,m). && (2.17)
\end{aligned}$$

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

Corollary 2.2.8. *When $k \nmid n$, $C_k(n,m) = nC_k(n-1,m)$ by Equation 2.3a, so*

$$\mathbb{E}[\pi(1) \mid \pi \in S_n \textit{ has exactly } m \textit{ k-cycles}] = \frac{n}{2}\left(1 - \frac{1}{n}\right) + 1 = \frac{n+1}{2}. \qquad (2.18)$$

Together with Theorem 2.2.1, this theorem and its corollary provides our first formula for the expected value of $\pi(1)$ that performs exponentially better than brute force.

2.2.4 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 2.2.9. *For $n < mk$ or $m < 0$, $C_k(n,m) = 0$. Otherwise, for all $k, m \geq 1$*

$$mC_k(n,m) = (k-1)!\binom{n}{k}C_k(n-k,m-1). \tag{2.19}$$

While this can be proven directly by the algebraic manipulation of the identity in Theorem 2.2.1, a bijective proof has been included here because it is natural and may be of interest.

Proof. Let

$$\mathscr{C}_k(n,m) = \{\pi \in S_n \mid \pi \text{ has exactly } m \text{ } k\text{-cycles}\}. \tag{2.20}$$

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

$$X^L_{n,m,k} = \{(\pi,c) \mid \pi \in \mathscr{C}_k(n,m), c \text{ a distinguished } k\text{-cycle of } \pi\}. \tag{2.21}$$

$$X^R_{n,m,k} = \{(\sigma,d) \mid \pi \in \mathscr{C}_k(n-k,m-1), d \text{ an } n\text{-ary necklace of length } k\}. \tag{2.22}$$

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:

$$(\pi_1\pi_2\cdots\pi_\ell, \pi_i) \overset{\varphi}{\mapsto} (\pi'_1\pi'_2\cdots\pi'_{i-1}\pi'_{i+1}\cdots\pi'_\ell, \pi_i) \tag{2.23}$$

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 $\#X^L_{n,m,k} = \#X^R_{n,m,k}$. $\square$

Example 2.2.10. *Suppose* $\pi = (423)(\mathbf{61})(75)$ *in cycle notation with* (61) *distinguished. Then*

$$\varphi((423)(61)(75),(61)) = ((312)(54),(61)) \tag{2.24}$$

under the bijection φ, described in the proof of Lemma 2.2.9.

The recurrence in Lemma 2.2.9 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 2.2.2 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 2.2.11. *For $k \geq 2$,*

$$\sum_{n=0}^{\infty} \frac{C_k(kn,0)k^n}{(kn)!}x^n = \frac{\exp(-x)}{1-kx}. \tag{2.25}$$

Proof. By substitution of $C_k(kn,0)$ via the identity in Theorem 2.2.1,

$$\begin{aligned}
\sum_{n=0}^{\infty} \frac{C_k(kn,0)k^n}{(kn)!}x^n &= \sum_{n=0}^{\infty}\sum_{i=0}^{n} \frac{(-1)^i}{k^i i!}k^n x^n &(2.26)\\
&= \sum_{n=0}^{\infty}\sum_{i=0}^{n} \frac{(-x)^i}{i!}(kx)^{n-i} &(2.27)\\
&= \left(\sum_{n=0}^{\infty} \frac{(-x)^n}{n!}\right)\left(\sum_{n=0}^{\infty}(kx)^n\right) &(2.28)\\
&= \frac{\exp(-x)}{1-kx}. &(2.29)
\end{aligned}$$

$\square$

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 2.1 unexplained. The following section will explain the connection between the expected values of $\pi(1)$ and the facet-derangements of the hypercube.

2.3 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.

2.3.1 Derangements of the generalized symmetric group

Definition 2.3.1. *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 2.3.2. *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:*

$$\underbrace{((1,3,0),(23))}_{\in S(k,n)} \cdot (0,1,1) = \underbrace{(23)}_{\in S_3} \cdot (1+0,3+1,0+1) = (23)\cdot(1,0,1) = (1,1,0). \tag{2.30}$$

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 2.1: 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 2.3.3. *A* ***derangement*** *or* ***fixed-point-free element*** *of the generalized symmetric group is an element* $((x_1,\dots,x_n),\pi) \in S(k,n)$ *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 2.3.4. *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 2.3.5 ([21])**.** *For* $k > 1$, *the number of derangements of the generalized symmetric group* $S(k,n)$ *is*

$$D(k,n) = k^n n! \sum_{i=0}^{n} \frac{(-1)^i}{k^i i!}. \tag{2.31}$$

which has exponential generating function

$$\sum_{n=0}^{\infty} \frac{D(k,n)}{n!} x^n = \frac{\exp(-x)}{1-kx}. \tag{2.32}$$

Notice that this agrees identically with the generating function in Lemma 2.2.11, which is our first hint in explaining the connection between k-cycles in permutations and fixed points in elements of the generalized symmetric group.

2.3.2 Permutation cycles and derangements

Lemma 2.3.6. *For $k \geq 1$, the number of permutations with $kn+km$ letters and m k-cycles is*

$$C_k(k(n+m),m) = \binom{kn+km}{kn} C_k(kn,0) \frac{(km)!}{k^m m!}. \tag{2.33}$$

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

$$C_k(k(n+m-1),m-1) = \frac{(km-k)!}{k^{m-1}(m-1)!} \binom{kn+km-k}{kn} C_k(kn,0). \tag{2.34}$$

$$= \frac{(kn+km-k)!}{k^{m-1}(m-1)!(kn)!} C_k(kn,0). \tag{2.35}$$

Rearranging Lemma 2.2.9,

$$C_k(k(n+m),m) = \frac{(k-1)!}{m} \binom{k(n+m)}{k} C_k(k(n+m-1),m-1) \tag{2.36}$$

$$= \frac{(kn+km)!}{km(kn+km-k)!} C_k(k(n+m-1),m-1). \tag{2.37}$$

Now, notice there is a $(kn+km-k)!$ term in the numerator of Equation 2.35 and the denominator of Equation 2.37, so substituting and simplifying yields

$$C_k(k(n+m),m) = \frac{(kn+km)!}{k^m m!(kn)!} C_k(kn,0), \tag{2.38}$$

as desired. $\square$

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. $\square$

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 2.3.7. *For $k \geq 2$,*

$$\frac{C_k(kn,m)}{(kn)!} = \binom{n}{m} \frac{D(k,n-m)}{k^n n!}. \tag{2.39}$$

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 2.3.5) is also the exponential generating function for

$$C_k(kn+km,m)\frac{m!n!k^{n+m}}{(kn+km)!}. \tag{2.40}$$

By the identity in Lemma 2.3.6,

$$\sum_{n=0}^{\infty} C_k(kn+km,m)\frac{m!n!k^{n+m}}{(kn+km)!}\frac{x^n}{n!} \tag{2.41}$$
$$= \sum_{n=0}^{\infty} \frac{(km)!}{m!k^m}\binom{kn+km}{kn} C_k(kn,0)\frac{m!n!k^{n+m}}{(kn+km)!}\frac{x^n}{n!} \tag{2.42}$$
$$= \sum_{n=0}^{\infty} C_k(kn,0)\frac{k^n x^n}{(kn)!} \tag{2.43}$$
$$= \frac{\exp(-x)}{1-kx}, \tag{2.44}$$

with the final equality being the identity in Lemma 2.2.11. □

2.3.3 Expected value of letters of permutations

We now have the ingredients we need to prove the pattern that we observed in Figure 2.1 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 2.3.8. *The expected value of the first letter of a permutation $\pi \in S_{kn}$ with exactly m k-cycles, where $k > 1$ and $0 \leq m \leq n$, is*

$$\mathbb{E}[\pi(1)\,|\,\pi \in S_{kn} \text{ has exactly } m \text{ } k\text{-cycles}] = \frac{kn+1}{2} + \frac{(-1)^{n-m}}{2D(k,n-m)} \tag{2.45}$$

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 2.3.7, yields

$$\frac{\frac{(kn)!}{n!k^n}\binom{n}{m}}{C_k(kn,m)} = \frac{1}{D(k,n-m)}. \tag{2.46}$$

Multiplying through by $(-1)^{n-m}$ to match the right hand side of Equation 2.45, together with some small manipulations yields

$$1-\frac{C_k(kn,m)-(-1)^{n-m}\frac{(kn)!}{n!k^n}\binom{n}{m}}{C_k(kn,m)}=\frac{(-1)^{n-m}}{D(k,n-m)}. \tag{2.47}$$

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

$$\begin{aligned}\frac{kn}{2}&\left(1-\frac{C_k(kn,m)-(-1)^{n-m}\frac{(kn)!}{n!k^n}\binom{n}{m}}{knC_k(kn,m)}\right)+1\\&=\frac{kn+1}{2}+\frac{(-1)^{n-m}}{2D(k,n-m)},\end{aligned} \tag{2.48}$$

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 2.2.2, the proof then follows from by Theorem 2.2.7. □

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 2.3.9. *The expected value of the i-th letter of a permutation in S_{kn} with exactly m k-cycles, where $n \in \mathbb{N}_{>0}$, $k > 1$, $1 \le i \le kn$, and $0 \le m \le n$, 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 2.2.5.

Thus

$$
\begin{aligned}
&\mathbb{E}[\pi(1) \,|\, \pi \in S_{kn} \text{ has exactly } m \text{ } k\text{-cycles}] \\
&\qquad = \frac{1}{N+(kn-1)M}\left(N+\sum_{a=2}^{kn} aM\right) \\
&\qquad = \frac{1}{N+(kn-1)M}\left(N+\left(\frac{kn(kn+1)}{2}-1\right)M\right). \qquad (2.49)
\end{aligned}
$$

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

$$
\begin{aligned}
&\mathbb{E}[\pi(i) \,|\, \pi \in S_{kn} \text{ has exactly } m \text{ } k\text{-cycles}] \\
&\qquad = \frac{1}{N+(kn-1)M}\left(N+\sum_{a\neq i} aM\right) \\
&\qquad = \frac{1}{N+(kn-1)M}\left(iN+\left(\frac{kn(kn+1)}{2}-i\right)M\right). \qquad (2.50)
\end{aligned}
$$

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 2.50, f is affine function in i. By Theorem 2.3.8, 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 2.3.10. *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 2.1.2. This agrees with Theorem 2.3.8:*

$$\frac{kn+1}{2} + \frac{(-1)^{n-m}}{2D(k,n-m)} = \frac{4+1}{2} + \frac{(-1)^{2-0}}{2D(2,2-0)} = \frac{5}{2} + \frac{1}{10} = \frac{13}{5}, \tag{2.51}$$

since $D(2,2) = 5$ as illustrated in Figure 2.1.

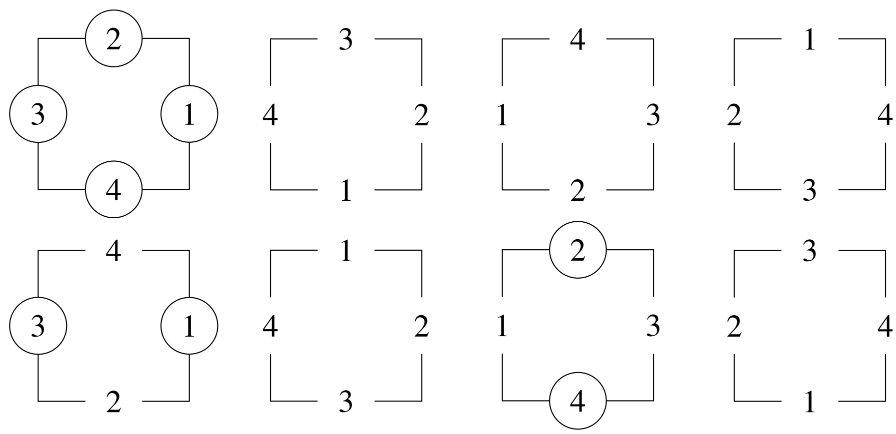

Figure 2.1: The $2^2 2! = 8$ symmetries of a square with fixed sides circled. The square (2-dimensional hypercube) has symmetry group $S(2,2) = (\mathbb{Z}/2\mathbb{Z}) \wr \mathbb{S}_2$ and $D(2,2) = 5$ of these symmetries are derangements, meaning that they do not fix any sides.

While Theorem 2.2.7 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 2.3.8 by using the formulas for $D(k,n)$ in Theorem 2.3.5. But this is not the only reason that Theorem 2.3.8 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 2.3.8 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 2.45 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.

2.4 A k-cycle preserving bijection

Motivated by Equation 2.3a, 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.

2.4.1 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 2.4.1. *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 $1 < 2 < \cdots < 9 < A < B < \cdots < N$.)*

$$\phi_3((D76)(E\sqcup)(F\underline{3}2\sqcup)(G\underline{9}1\underline{C})(K\sqcup 5\underline{4})(L\sqcup J\underline{8})(M\sqcup B\sqcup)(N\underline{A}H\sqcup),\underline{I})$$

$$= (D76)(E3)(F29)(G1)(KC5)(L4J)(M8BA)(NHI)$$

$$\psi_3 \qquad \psi_3 \qquad \phi_3 \qquad \phi_3 \qquad \phi_3 \qquad \psi_3 \qquad \psi_3$$

$$\begin{aligned}
&\psi_3((D76)(E\underline{3})(F\textvisiblespace 2\underline{9})(G\textvisiblespace 1\textvisiblespace)(K\underline{C}5\textvisiblespace)(L\underline{4}J\textvisiblespace)(M\underline{8}B\underline{A})(N\textvisiblespace H\underline{I}))\textvisiblespace \\
&\quad = ((D76)(E)(F32)(G91C)(K54)(LJ8)(MB)(NAH), I)
\end{aligned}$$

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.

2.4.2 Formal definition and properties

$$\phi_3 \qquad \phi_3 \qquad \psi_3 \qquad \psi_3 \qquad \psi_3 \qquad \phi_3 \qquad \phi_3$$

Definition 2.4.2. *Define $\phi_k \colon S_{n-1} \times [n] \mapsto S_n$ recursively as follows:*

$$\phi_k(\emptyset, 1) = (1), \tag{2.52}$$

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

$$\phi_k(\pi, x) = \begin{cases} c^{(t)} \cdots c^{(1)}(x) & x > c_1^{(1)} & (2.53a) \\ \phi_k(c^{(t)} \cdots c^{(2)}, c_2^{(1)})(c_1^{(1)} c_3^{(1)} \cdots c_k^{(1)} x) & \ell_1 = k & (2.53b) \\ \pi'(c_1^{(1)} x' c_2^{(1)} \cdots c_{k-1}^{(1)} x) & \ell_1 = k-1, t > 1 & (2.53c) \\ c^{(t)} \cdots c^{(2)}(c_1^{(1)} \cdots c_{\ell_1}^{(1)} x) & \textit{otherwise.} & (2.53d) \end{cases}$$

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

$$\psi_k(\pi) = \begin{cases} \left(c^{(t)} \ldots c^{(2)}, c_1^{(1)}\right) & \ell_1 = 1 & (2.54a) \\ \left(\phi_k(c^{(t)} \cdots c^{(2)}, a_2^{(1)})(c_1^{(1)} c_3^{(1)} \ldots c_k^{(1)}), c_{k+1}^{(1)}\right) & \ell_1 = k+1 & (2.54b) \\ \left(\pi'(c_1^{(1)} x' c_2^{(1)} \ldots c_{k-1}^{(1)}), c_k^{(1)}\right) & \ell_1 = k, t > 1 & (2.54c) \\ \left(c^{(t)} \cdots c^{(2)}(c_1^{(1)} \cdots c_{\ell_1 - 1}^{(1)}), c_{\ell_1}^{(1)}\right) & \textit{otherwise,} & (2.54d) \end{cases}$$

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

Note 2.4.3. *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 2.4.7, it implies Equation 2.3a.

Theorem 2.4.4. *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 2.4.5. *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 2.53b, 2.53c, 2.54b, or 2.54c), which we look at case-by-case.

Case 1. In Equation 2.53b, 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 2.53c, 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 2.54b, 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 2.54c, 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 2.4.6. *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 2.53a 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 2.54a, but in this case, the cycle only has one letter, so this is equivalent to deleting the cycle. □

2.4.3 Inverting the bijection

Lemma 2.4.7. *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 2.53a:

$$\psi_k(\phi_k(\pi,x)) = \psi_k(c^{(t)} \cdots c^{(1)}(x)) \tag{2.55}$$

$$= (c^{(t)} \cdots c^{(1)}, x) \tag{2.56}$$

$$= (\pi, x). \tag{2.57}$$

Case 2. Assume $\ell_1 = k$, so that $\phi_k(\pi, x)$ is evaluated via Equation 2.53b:

$$\psi_k(\phi_k(\pi,x)) = \psi_k(\phi_k(c^{(t)} \cdots c^{(2)}, c_2^{(1)})\underbrace{(c_1^{(1)} c_3^{(1)} \cdots c_k^{(1)} x)}_{\text{length } k}) \tag{2.58}$$

$$= (\pi'(c_1^{(1)} x' c_3^{(1)} \ldots c_k^{(1)}), x) \tag{2.59}$$

Case 3. Assume $\ell_1 = k-1$ and $t > 1$, so that $\phi_k(\pi, x)$ is evaluated via Equation 2.53c:

$$\psi_k(\phi_k(\pi,x)) = \psi_k(\pi' \underbrace{(c_1^{(1)} x' c_2^{(1)} \cdots c_{k-1}^{(1)} x)}_{\text{length } k+1}) \tag{2.60}$$

where $(\pi', x') = \psi_k(c^{(t)} \ldots c^{(2)})$. Therefore, this simplifies by Equation 2.54c:

$$\psi_k(\phi_k(\pi,x)) = \left(\phi_k(\pi',x')(c_1^{(1)} \cdots c_{k-1}^{(1)}), x\right) \tag{2.61}$$

$$= \left(\underbrace{\phi_k(\psi_k(c^{(t)} \ldots c^{(2)}))}_{c^{(t)} \ldots c^{(2)}} \underbrace{(c_1^{(1)} \cdots c_{k-1}^{(1)})}_{c^{(1)}}, x\right) \tag{2.62}$$

$$= (\pi, x), \tag{2.63}$$

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 2.53d:

$$\psi_k(\phi_k(\pi,x)) = \psi_k(c^{(t)} \cdots c^{(2)} (c_1^{(1)} \cdots c_{\ell_1}^{(1)} x)) \tag{2.64}$$

$$= (c^{(t)} \cdots c^{(1)}, x) \tag{2.65}$$

$$= (\pi, x). \tag{2.66}$$

Case 5. Assume that $\ell_1 = k-1$ and $t = 1$, so that $\phi_k(\pi, x)$ is evaluated via Equation 2.53d:

$$\psi_k(\phi_k(\pi,x)) = \psi_k((c_1^{(1)} \cdots c_{k-1}^{(1)} x)) \tag{2.67}$$

$$= (c^{(1)}, x) \tag{2.68}$$

$$= (\pi, x). \tag{2.69}$$

$\square$

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.

2.5 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 $\mathrm{stat}\colon S_n \to \mathbb{Z}$, does the map

$$f(n,m) = \mathbb{E}[\pi(i) \mid \pi \in S_n, \mathrm{stat}(\pi) = m] \tag{2.70}$$

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 $(\mathrm{stat}_1, \mathrm{stat}_2)$, does the map

$$g(n,m) = \mathbb{E}[\mathrm{stat}_1(\pi) \mid \pi \in S_n, \mathrm{stat}_2(\pi) = m] \tag{2.71}$$

have any interesting structure?

2.5.1 FindStat database

The result by Conger gives the expected value of $\pi(1)$ given $\mathrm{des}(\pi)$, 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 [22] 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.

2.5.2 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.[23] 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 2.5.1. *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 2.5.2. *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,*

$$\mathbb{E}[\pi(1) \mid \pi \in S_n, \mathrm{maj}(\pi) = 1] = \frac{1}{2}\left(\frac{n^2+n-2}{n-1}\right), \tag{2.72}$$

$$\mathbb{E}[\pi(1) \mid \pi \in S_n, \mathrm{maj}(\pi) = 2] = \frac{1}{3}\left(\frac{n^3-n-6}{n^2-n-2}\right), \tag{2.73}$$

$$\mathbb{E}[\pi(1) \mid \pi \in S_n, \mathrm{maj}(\pi) = 3] = \frac{1}{4}\left(\frac{n^4+6n^3-13n^2-18n}{n^3-7n}\right), \textit{ and} \tag{2.74}$$

$$\mathbb{E}[\pi(1) \mid \pi \in S_n, \mathrm{maj}(\pi) = 4] = \frac{1}{5}\left(\frac{n^5+20n^4-45n^3-80n^2-16n}{n^4+2n^3-13n^2-14n}\right). \tag{2.75}$$

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

2.5.3 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 2.3.7 suggests, there exists a bijection of sets

$$C_k(n,\alpha) \times F_k(n,\beta) \to C_k(n,\beta) \times F_k(n,\alpha). \tag{2.76}$$

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 3

Deranking Menage

TODO

3.1 TODO

1. Introduction
2. We can also *rank* a given permutation
3. Get ListOfTables working and captions working for tables.
4. Define **derived** complementary board B_α^c?
5. If we do a cyclic rotation of the rows of a chessboard, we get essentially the same thing.
6. Move code to Appendix.
7. Define B_α and $\overline{B}_\alpha^c$.
8. Do we want to talk about parking functions?
9. Is it worthwhile to discuss prefix functions for compositions, etc.?
10. Decide on "derank" or "unrank"

3.2 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?

Definition 3.2.1. *Let $\mathscr{C}$ be a totally ordered finite set, and let $\{c_i\}_{i=1}^{|\mathscr{C}|}$ be the unique sequence of elements in $\mathscr{C}$ such that $c_i < c_{i+1}$ for all $1 \leq i < |\mathscr{C}|$.*

*The **ranking map** is the map $\operatorname{rank}_{\mathscr{C}} : \mathscr{C} \to \mathbb{N}_{>0}$ which sends $c_i \mapsto i$.*

*The **deranking map** is the inverse map $\operatorname{derank}_{\mathscr{C}} : \mathbb{N}_{>0} \to \mathscr{C}$ which sends $i \mapsto c_i$.*

In abstract terms, these maps are not particularly interesting, but in practical terms it can be quite difficult to efficiently compute a given ranking or deranking from a given totally ordered set. After all, when these sets grow in exponential time or worse, explicitly constructing the sequence and doing a search is not computationally feasible.

In Appendix TODO, we will provide examples of total orders on combinatorial objects $\mathscr{C}$ for which constructing efficient

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 [24] (de Bruijn Sequences?), Dyck path in lexicographic order?

3.3 Prefix Counting and Word Ranking

Lemma 3.3.1. *If we have an efficient way to compute the deranking map, an efficient way to compare two elements in the total order, and an efficient way of computing the number of objects at hand, $|\mathscr{C}|$, then we can efficiently compute the ranking map.*

Proof. We can do a binary search. (TODO: write pseudo-code algorithm?) □

3.3.1 Counting Words With a Given Prefix

In both the case of deranking derangements and menage permutations (and in many other applications) our combinatorial objects are words and our total order is lexicographic order.

Definition 3.3.2. ***Lexicographic order*** *is a total ordering on words where $w < v$ ... TODO*

Definition 3.3.3. *A finite* **word** *w over an alphabet $\mathscr{A}$ is a finite sequence $\{w_i \in \mathscr{A}\}_{i=1}^N$.*

The collection of finite words over the alphabet $\mathscr{A}$ is denoted by $\mathscr{W}_{\mathscr{A}}$, or just $\mathscr{W}$ when the alphabet is implicit from context.

Definition 3.3.4. *A word $w = \{w_i \in \mathscr{A}\}_{i=1}^N$ is said to begin with a* ***prefix*** *$\alpha = \{\alpha_i \in \mathscr{A}\}_{i=1}^M$ if $M \leq N$ and $w_i = \alpha_i$ for all $i \leq M$.*

Lemma 3.3.5. *Let $\mathscr{W}$ be the set of words of any length on the alphabet $[n]$, and let $\mathscr{C} \subsetneq \mathscr{W}$ be a finite subset of words on this alphabet, with a total order equal to its lexicographic order.*

Then let $\#\text{prefix}_{\mathscr{C}} : \mathscr{W} \to \mathscr{C}$ be the function that counts the number of words in $\mathscr{C}$ that begin with a given prefix.

Then the deranking function can be computed recursively by

$$\text{derank}_{\mathscr{C}}(i) = f_i^{\mathscr{C}}((1), 0)$$

where

$$f_i^{\mathscr{C}}(\alpha, j) = \begin{cases} \alpha & i \in (j, j + \#\mathrm{prefix}_{\mathscr{C}}(\alpha)] \textit{ and } \alpha \in \mathscr{C} \\ f_i^{\mathscr{C}}(\alpha', j) & i \in (j, j + \#\mathrm{prefix}_{\mathscr{C}}(\alpha)] \textit{ and } \alpha \notin \mathscr{C} \\ f_i^{\mathscr{C}}(\alpha'', j + \#\mathrm{prefix}_{\mathscr{C}}(\alpha)) & \textit{otherwise}, \end{cases} \tag{3.1}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$, $\alpha' = (\alpha_1, \alpha_2, \ldots, \alpha_\ell, 1)$, $\alpha'' = (\alpha_1, \alpha_2, \ldots, \alpha_{\ell-1}, 1 + \alpha_\ell)$, *and* j *denotes the number of words in* $\mathscr{C}$ *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. □

This shows that if we can construct a function $\#\mathrm{prefix}_{\mathscr{C}}$ that efficiently counts the number of elements of $\mathscr{C}$ with a given prefix, then we can efficiently rank and derank the elements of $\mathscr{C}$ in lexicographic order.

3.3.2 Ranking words

In Lemma 3.3.1, we showed that given an efficient algorithm to compute $\mathrm{derank}_{\mathscr{C}}$, we can derive an efficient algorithm to compute $\mathrm{rank}_{\mathscr{C}}$, on the order of $O(\log(|\mathscr{C}|\,\mathrm{derank}_{\mathscr{C}}(n)))$ (TODO: make the size of the input explicit.)

However, we can provide a faster algorithm via another recursive function: $\mathrm{rank}_{\mathscr{C}}(w) = g_w(1,1,0)$ where

$$g_w(i, \ell, c) = \begin{cases} c + 1 & \ell = w_i \text{ and } i = |w| \\ g_w(i+1, 1, c) & \ell = w_i \text{ and } i < |w| \\ g_w(i, \ell+1, c + \#\mathrm{prefix}_{\mathscr{C}}(w')) & \ell < w_i, \end{cases} \tag{3.2}$$

where $w' = (w_1, w_2, \ldots, w_{i-1}, \ell)$.

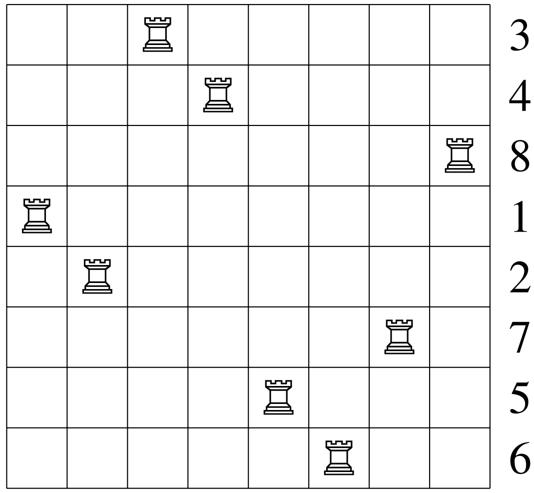

Figure 3.1: An illustration of the rook placement corresponding to the permutation $34812756 \in S_8$. A rook is placed in square $(i, \pi(i))$ for each i.

3.3.3 Basic Notions of Rook Theory

Because we have showed that we can ... In the case of deranking derangements and permutations, it is useful to use ideas from Rook Theory. Rook Theory was introduced by Kaplansky [25] Riordan [26] 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.

Definition 3.3.6. *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 3.3.7. *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 [27] claims that "the theory of enumerating the permutations with restricted positions stimulated the development of the theory of the permanent.")

Lemma 3.3.8. *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

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

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

Theorem 3.3.9 (Valiant's Theorem [28])**.** *Computing the permanent of a (0,1)-matrix is #P-complete.*

Corollary 3.3.10. *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.

3.3.4 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 3.3.11 ([26]). *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*

$$B_i = \{(x',y') \in B : x \neq x' \text{ and } y \neq y'\} \tag{3.3}$$

$$B_e = B \setminus (x,y). \tag{3.4}$$

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

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

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.

3.4 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 [29] 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 3.4.1. *A derangement is a permutation* $\pi \in S_n$ *such that* π *has no fixed points. That is, the set of derangements is*

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

3.4.1 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 3.2.

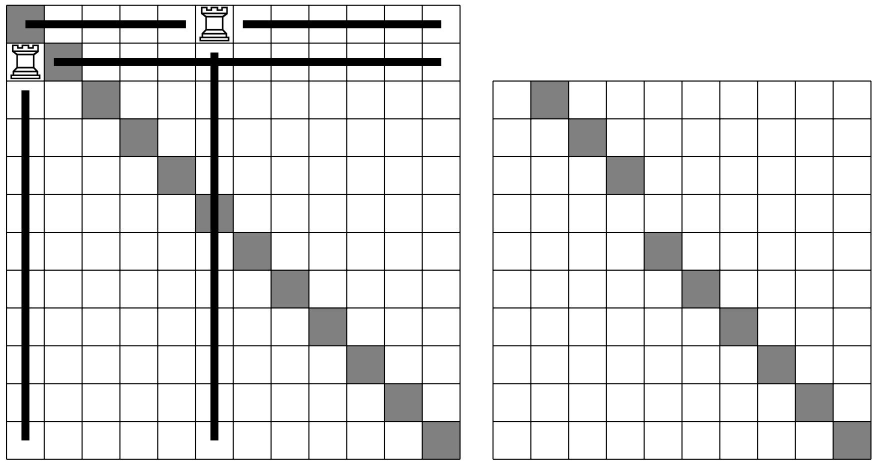

Figure 3.2: An example of a prefix $\alpha = (6,1)$, and the board that results from deleting the first $\ell = 2$ rows and columns 6 and 1. The derived complementary board of B from α is $B^c_\alpha = \{(1,2),(2,3),(3,4),(5,5),\ldots,(10,10)\}$.

Definition 3.4.2. *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 3.4.3. *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*

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

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

Proof. TODO $\square$

3.4.2 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 3.4.4. *The rook polynomial for the complementary board B^c_α is*

$$p_{B^c_\alpha}(x) = \sum_{j=0}^{|B^c_\alpha|} \binom{|B^c_\alpha|}{j} x^j. \tag{3.5}$$

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 $|B^c_\alpha|$. $\square$

Now we introduce a lemma of Stanley [30] to compute the number of TODO from a complementary board.

Lemma 3.4.5 ([30])**.** *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 3.4.6. *The number of derangements with prefix $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ 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 [10].*

Example 3.4.7. *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, $\{3, 4, \ldots, 14\} \cap \{6, 1\} = 6$. Thus $k = 12 - 1 = 11$. Thus there are* A047920$(12, 11) =$ $190\,899\,411$ *derangements that start with* $6\,1$.

3.5 Deranking Ménage Permutations

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

α (prefix)	#prefix(α)	index range	$\lvert B^c_\alpha \rvert$	$\mathrm{derank}_i(\alpha,\ell)$
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	$\mathrm{derank}_{1000}(25487361,999)$

Table 3.1: There are $A000166(8) = 14833$ derangements on 8 letters. This algorithm finds the derangement at index 1000.

Definition 3.5.1. *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 $\Rightarrow$ grouped columns $\Rightarrow$ partition/multiset $\Rightarrow$ complementary polynomial $\Rightarrow$ count

3.5.1 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 3.5.2. *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 3.5.3 ([25])**.** *If B can be partitioned into disjoint boards $b_1, b_2, \ldots, b_m$, 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 3.3.

Lemma 3.5.4. *For $\ell \geq 1$, and prefix $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ the derived complementary board B^c_α can be partitioned into boards of one of three shapes.*

1. *(TODO Figure 3.3, left)*

2. *(TODO Figure 3.3, middle)*

3. *(TODO Figure 3.3, right)*

Proof. The proof proceeds by induction. Base case: because of the ménage restriction, $\pi(1) \in \{2,3,\ldots,n-1\}$, 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 □

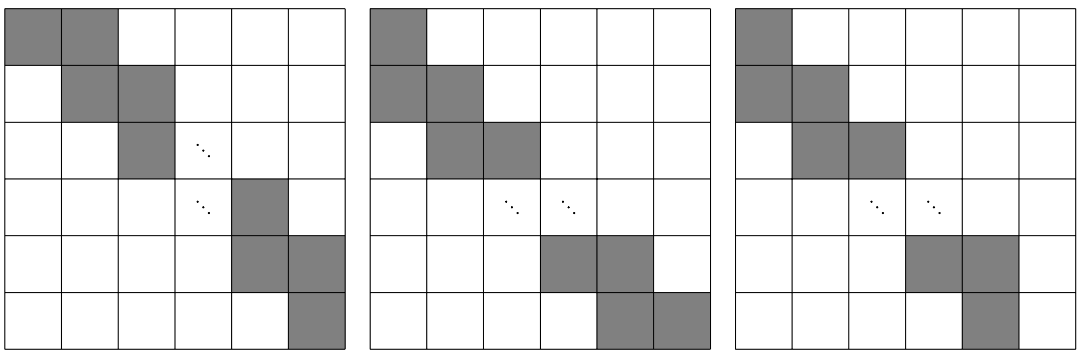

Figure 3.3: Three $n \times n$ blocks, two with $2n-1$ crossed-out cells and one with $2n-2$ crossed-out cells.

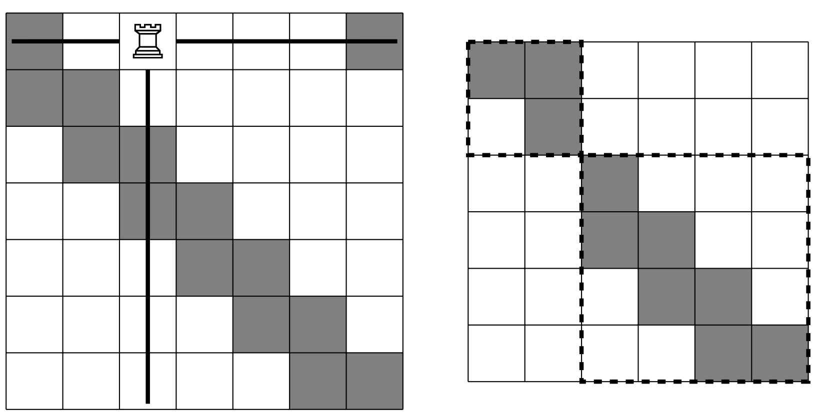

Figure 3.4: The first chessboard shows a placement of a rook at position 3, the second shows the remaining squares, and the third shows a permutation of the rows to put the board into a block-diagonal form.

3.5.2 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 [10].

Definition 3.5.5. *For $j \geq 0$, the jth* ***Fibonacci polynomial*** *$F_j(x)$ is defined recursively as:*

$$F_0(x) = 1 \tag{3.6}$$

$$F_1(x) = 1 + x \tag{3.7}$$

$$F_n(x) = F_{n-1}(x) + xF_{n-2}(x). \tag{3.8}$$

Lemma 3.5.6. *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 3.3.11, and proceed by induction on the upper-left square.

TODO (See Figure 3.3, 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.

□

3.5.3 Prefix to blocks

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

Lemma 3.5.7. *Given a prefix* $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ *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:*

$$c_i = \begin{cases} 0 & i < \ell \\ 1 & i = k \text{ or } i = n \\ 2 & \ell < i < n \end{cases} \tag{3.9}$$

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

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

Lemma 3.5.8. *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")*

3.5.4 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 sub-boards 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 3.5.9. *Given a board* B^c_α *that is partitioned into disjoint boards* $b_1, b_2, \ldots, b_m$*, 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 $p_{B^c_\alpha}$, 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.

3.5.5 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*

$$7\ 16\ 19\ 12\ 2\ 8\ 15\ 1\ 18\ 14\ 3\ 9\ 20\ 10\ 5\ 17\ 13\ 4\ 11\ 6. \tag{3.10}$$

Example 3.5.10. *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.*

3.6 Generalizations and Open Questions

3.6.1 Other restricted permutations

Doron Zeilberger considers a more general family of restricted permutations.

Definition 3.6.1 ([31])**.** *Let $S \subset \mathbb{Z}$, then a S-avoiding permutation is a permutation $\pi \in S_n$ such that*

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

Example 3.6.2. *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	$\mathrm{derank}_i(\alpha,\ell)$
1	0	$(0,0]$	–	$\mathrm{derank}_{1000}(1,0)$
2	787	$(0,787]$	$(1,11)$	$\mathrm{derank}_{1000}(2,0)$
3	791	$(787,1578]$	$(3,9)$	$\mathrm{derank}_{1000}(3,787)$
31	0	$(787,787]$	–	$\mathrm{derank}_{1000}(31,787)$
32	0	$(787,787]$	–	$\mathrm{derank}_{1000}(32,787)$
33	0	$(787,787]$	–	$\mathrm{derank}_{1000}(33,787)$
34	159	$(787,946]$	$(1,7)$	$\mathrm{derank}_{1000}(34,787)$
35	166	$(946,1112]$	$(1,2,5)$	$\mathrm{derank}_{1000}(35,946)$
351	24	$(946,970]$	$(0,2,5)$	$\mathrm{derank}_{1000}(351,946)$
$\cdots$	0	$(970,970]$	–	
354	34	$(970,1004]$	$(0,5)$	$\mathrm{derank}_{1000}(354,970)$
3541	5	$(970,975]$	$(0,5)$	$\mathrm{derank}_{1000}(3541,970)$
3542	5	$(975,980]$	$(0,5)$	$\mathrm{derank}_{1000}(3542,975)$
$\cdots$	0	$(980,980]$	–	
3546	8	$(980,988]$	$(0,3)$	$\mathrm{derank}_{1000}(3546,980)$
3547	10	$(988,998]$	$(0,2,1)$	$\mathrm{derank}_{1000}(3547,988)$
3548	6	$(998,1004]$	$(0,4)$	$\mathrm{derank}_{1000}(3548,998)$
35481	1	$(998,999]$	$(0,4)$	$\mathrm{derank}_{1000}(35481,998)$
35482	1	$(999,1000]$	$(0,4)$	$\mathrm{derank}_{1000}(35482,999)$
354821	0	$(999,999]$	(3)	$\mathrm{derank}_{1000}(354821,999)$
$\cdots$	0	$(999,999]$	–	
354827	1	$(999,1000]$	$(0,1)$	$\mathrm{derank}_{1000}(354827,999)$
3548271	1	$(999,1000]$	(0)	$\mathrm{derank}_{1000}(3548271,999)$
35482716	1	$(999,1000]$	$()$	$\mathrm{derank}_{1000}(35482716,999)$

Table 3.2: The recursive computation of the 1000th ménage permutation.

3.6.2 Observation about Lyndon Words after? a given prefix

Definition 3.6.3. *A Lyndon word is a string that is the unique minimum with respect to all of its rotations.*

Example 3.6.4. 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 3.6.5. *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*.*

References

1. Gardner, M. MATHEMATICAL GAMES. *Scientific American* **240,** 16–27 (1979).
2. Winkler, P. *Mathematical Puzzles: A Connoisseur's Collection* (AK Peters, Natick, Mass, 2004).
3. Gardner, M. MATHEMATICAL GAMES. *Scientific American* **240,** 21–31 (1979).
4. Yehuda, R. B., Etzion, T. & Moran, S. Rotating-Table Games and Derivatives of Words. *Theor. Comput. Sci.* **108,** 311–329 (1993).
5. Ehrenborg, R. & Skinner, C. M. The Blind Bartender's Problem. *Journal of Combinatorial Theory, Series A* **70,** 249–266. doi:`https://doi.org/10.1016/0097-3165(95)90092-6` (1995).
6. Roeder, O. The Riddler. *FiveThirtyEight* (2019).
7. Sidana, T. *Constacyclic codes over finite commutative chain rings* PhD thesis (Indraprastha Institute of Information Technology, Delhi, 2020).
8. Rabinovich, Y. A generalization of the Blind Rotating Table game. *Information Processing Letters* **176,** 106233. doi:`https://doi.org/10.1016/j.ipl.2021.106233` (2022).
9. Rotman, J. J. *An Introduction to the Theory of Groups* (Springer, New York, 1999).
10. Inc., O. F. *The On-Line Encyclopedia of Integer Sequences* 2021.
11. Mazurov, V. D. On Generation of Sporadic Simple Groups by Three Involutions Two of Which Commute. *Siberian Mathematical Journal* **44,** 160–164. doi:`10.1023/A:1022028807652` (2003).
12. Nuzhin, Y. N. Generating triples of involutions of alternating groups. *Mathematical Notes* **51,** 389–392. doi:`10.1007/BF01250552` (1992).
13. Nuzhin, Y. N. Generating triples of involutions of Chevalley groups over a finite field of characteristic 2. *Algebra i Logika* **29,** 192–206, 261. doi:`10.1007/BF02001358` (1990).
14. Nuzhin, Y. N. Generating triples of involutions of Lie-type groups over a finite field of odd characteristic. I. *Algebra i Logika* **36,** 77–96, 118. doi:`10.1007/BF02671953` (1997).
15. Nuzhin, Y. N. Generating triples of involutions of Lie-type groups over a finite field of odd characteristic. II. *Algebra i Logika* **36,** 422–440, 479 (1997).
16. (https://math.stackexchange.com/users/155629/travis-willse), T. W. *The unique loop (quasi-group with unit) L of order* 5 *satisfying* $x^2 = 1$ *for all* $x \in L$ Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/1216907 (version: 2018-12-27).
17. Winkler, P. *Mathematical Puzzles* (CRC Press, 2021).
18. 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).
19. V. Goncharov. Du domaine de l'analyse combinatoire. *Izv. Akad. Nauk SSSR Ser. Mat.* **8,** 3–48 (1 1944).

20. Arratia, R. & Tavare, S. The Cycle Structure of Random Permutations. *The Annals of Probability* **20,** 1567–1591 (1992).
21. Assaf, S. H. Cyclic Derangements. *The Electronic Journal of Combinatorics* **17** (2010).
22. Rubey, M., Stump, C., *et al. FindStat - The combinatorial statistics database* `http://www.FindStat.org`. Accessed: May 22, 2022.
23. Foata, D. *Distributions Euleriennes et Mahoniennes sur le Groupe des Permutations* in *Higher Combinatorics* (ed Aigner, M.) (Springer Netherlands, Dordrecht, 1977), 27–49.
24. 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.
25. 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).
26. Riordan, J. *An Introduction to Combinatorial Analysis* (Princeton University Press, USA, 1980).
27. 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).
28. 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).
29. Mikawa, K. & Tanaka, K. Lexicographic ranking and unranking of derangements in cycle notation. *Discret. Appl. Math.* **166,** 164–169 (2014).
30. Stanley, R. P. *Enumerative Combinatorics: Volume 1* 2nd (Cambridge University Press, USA, 2011).
31. Zeilberger, D. Automatic Enumeration of Generalized Ménage Numbers. *Séminaire Lotharingien de Combinatoire* **71** (2014).