An Exploration of Pattern Avoidance in Permutations

Dallas Anderson

Dec 7 2025

Abstract

This paper is about pattern avoidance in permutations. In it, we will explore concepts such as Wilf-equivalence, enumerate permutations of a given length avoiding a given pattern, and prove an unexpected result connecting pattern avoidance to stack-sortability. At the end, we will briefly touch on other notions of pattern avoidance in combinatorial structures outside of permutations.

1 Introduction

Pattern avoidance in permutations is a fascinating combinatorial subject. Countless work is being done in the field, with many questions still unanswered. But before we get into the open problems, we need to learn some of the basics. Let's start with some notation:

Definition 1.1. We write $[n]$ for the set of all integers k such that $1 \leq k \leq n$.

Definition 1.2. Take $\sigma_1, \ldots, \sigma_n \in [n]$. We write $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ for the function $\sigma : [n] \to [n]$ such that $\sigma(k) = \sigma_k$ for all $k \in [n]$, and we say that σ has *length* n. Typically, σ will be a permutation.

Next, we have to define order-isomorphism of sequences.

Definition 1.3. Take two sequences $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ of numbers. We say that the sequences are *order-isomorphic* when $a_i < a_j$ if and only if $b_i < b_j$ for all $i, j \leq n$.

For example, the sequences $1, 3, 4$ and $2, 6, 7$ are order-isomorphic because they're both in strictly increasing order. Also, the sequence $4, 8, 5$ is order-isomorphic to $1, 3, 2$ because they both go in the order of smallest number, biggest number, middle number. However, the sequences $1, 3, 2, 4$ and $8, 6, 7, 9$ are not order-isomorphic; for instance, the first sequence starts with its smallest value whereas the second one doesn't. Now, let's look at pattern avoidance.

Definition 1.4. Let $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ and $\pi = \pi_1\pi_2\cdots\pi_m$ be permutations on $[n]$ and $[m]$ respectively. If there's some subsequence of $\sigma_1, \ldots, \sigma_n$ that's order-isomorphic to the sequence $\pi_1, \ldots, \pi_m$, then we say that π *is a pattern of* σ, or that σ *contains* π *as a pattern.* If σ does not contain π, we say that σ *avoids* π. We denote the fact that π is a pattern of σ by writing $\pi \preceq \sigma$. If additionally, $\pi \neq \sigma$, we write $\pi \prec \sigma$.

Remark 1.5. In this paper, when we write something like 1324, it will usually indicate a permutation on $[4]$, not the number $1,324$. Whenever we use multiple-digit numbers, which we won't often, it should be clear from context.

Now, an interesting question that's being studied is how many permutations avoid a given pattern. More precisely, given a permutation π, we want to know how many permutations $\sigma : [n] \to [n]$ avoid π for a given n. For example, let π be the permutation 123. It has been shown that the number of permutations of $[n]$ that avoid π is the nth Catalan number, C_n. [6] Let's take another permutation, say 132. As it turns out the number of permutations of $[n]$ that avoid 132 is also C_n! [6] Such equivalences are surprisingly common. They're so common, in fact, that they've been given a name.

Definition 1.6. Take any two permutations σ and τ. We say that σ and τ are *Wilf-equivalent* if, for every n, the number of permutations of $[n]$ that avoid σ equals the number of permutations of $[n]$ that avoid τ.

Now, there are some Wilf-equivalences that are obvious. For example, the permutation $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ of $[n]$ is Wilf-equivalent to $\tau = \sigma_n\sigma_{n-1}\cdots\sigma_1$, because the permutations that avoid τ are just the reverse of the permutations that avoid σ. Similarly,

$$\pi = (n+1-\sigma_1)\cdots(n+1-\sigma_n)$$

is Wilf-equivalent to σ, and σ^{-1}, the inverse function of σ, is also Wilf-equivalent to σ. However, the equivalence between 132 and 123 takes a bit more work to prove.

2 Enumerating Permutations Avoiding a Given Pattern

From the fact that 132 and 123 are Wilf-equivalent, one can use reverses and complements to show that, in fact, all patterns of length 3 are Wilf-equivalent.

Theorem 2.1. All permutation patterns of length 3 are Wilf-equivalent. [6]

We won't prove that theorem here, but we will show that the number of patterns of length n avoiding 132 is C_n. This combined with Theorem 2.1 shows that the same is true for all patterns of length 3. Before we prove that, though, we will need some notation:

Definition 2.2. Let σ be a permutation of length $n \geq 1$. We write $S_\sigma(k)$ for the set of all permutations of length k that avoid σ, and we write $s_\sigma(k)$ for the number of elements in $S_\sigma(k)$.

Theorem 2.3. For all $n \geq 1$, $s_{132}(n) = C_n$. [6]

Proof. Take any permutation σ that avoids 132, and suppose that the element n is in the ith entry. Note that if x is an element before n and y is after n so that $x < y$, then xny is a containment of 132, a contradiction. Thus everything to the left of n is larger than everything to the right of n. This in particular means that, since n is in slot i, the elements to the left of n are a rearrangement of the numbers $n-i+1 \leq k \leq n-1$, and the elements to the right of n are a rearrangement of the numbers $1 \leq k \leq n-i$. The former sequence is thus a 132-avoiding permutation of $[i-1]$ shifted upward, and the latter is a 132-avoiding permutation of $[n-i]$. So the amount of possible σ that can be constructed with n in the ith place is $s_{132}(i-1)s_{132}(n-i)$. Summing over i, we get

$$s_{132}(n) = \sum_{i=1}^{n} s_{132}(i-1)s_{132}(n-i).$$

Notice that one of the s terms in the product could sometimes be $s_{132}(0)$. In those cases, the permutation is empty on that side, and since there is only 1 empty string, we may write $s_{132}(0) = 1$. This is the same initial condition and recurrence relation as the Catalan numbers C_n, so the two sequences of numbers must be the same. □

These two theorems raise an important question: Given any pattern π, can you find a nice way of expressing the function $s_\pi(n)$? Perhaps there's some formula that can evaluate $s_\pi(n)$ for any π, n. The cases where π has length 3 are particularly nice cases, since $s_\pi(n)$ has a neat formula:

$$s_\pi(n) = C_n = \frac{1}{2n+1}\binom{2n+1}{n}.$$

This is the formula for the Catalan numbers.

In the general case, however, there isn't such a neat formula. For example, let's look at the permutations of length 4. First of all, there are 3 Wilf-equivalence classes of the permutations of length 4. One of them has a representative 1342, and the others have representatives 1234 and 1324 respectively. [9] The equivalence class [1342] does have a formula for its s sequence, but it's significantly messier than the formula for $s_{132}(n)$.

Theorem 2.4. [4]

$$s_{1342}(n) = \frac{(7n^2-3n-2)}{2}\cdot(-1)^{n-1} + 3\sum_{i=2}^{n} 2^{i+1}\cdot\frac{(2i-4)!}{i!(i-2)!}\cdot\binom{n-i+2}{2}\cdot(-1)^{n-i}.$$

Additionally, the formula for $s_{132}(n)$ is in *closed form*; it has the same amount and structure of terms regardless of n. However, because the formula for $s_{1342}(n)$ has a summation in it, the number of terms in the expression depends on n. This is not as convenient, because the bigger n is, the more time it would take to calculate $s_{1342}(n)$. However, one nice thing is that we can use this formula to derive an *asymptotic* for $s_{1342}(n)$:

Theorem 2.5. The smallest c for which $s_{1342} < c^n$ for all n is $c = 8$. [4]

Now let's look at the class [1234]. Just as with 1342, there's a formula for $s_{1234}(n)$:

Theorem 2.6. [5]

$$s_{1234}(n) = \frac{1}{(n+1)^2(n+2)}\sum_{k=0}^{n}\binom{2k}{k}\binom{n+1}{k+1}\binom{n+2}{k+1}.$$

This formula may look a little cleaner than the formula for $s_{1342}(n)$, but it still presents the same issue as before: It doesn't have a closed form. Just like s_{1342}, this one does have an asymptotic, but in a different form:

Theorem 2.7. There is a constant c such that

$$\lim_{n\to\infty}\frac{s_{1234}(n)}{c\cdot\frac{9^n}{n^4}} = 1.$$

In other words, $s_{1234}(n)$ is *asymptotically equal* to $c\cdot\frac{9^n}{n^4}$ as $n\to\infty$.

This is a specific case of a more general theorem by [10]:

Theorem 2.8. For all k, $s_{123\cdots k}(n)$ is asymptotically equal to

$$c\frac{(k-1)^{2n}}{n^{\frac{k^2-2k}{2}}}$$

for some constant c.

One thing to note is that it isn't even trivial that these s functions have exponential growth. For example, imagine if there were some permutation π such that for large n, only a very small fraction of the permutations on $[n]$ *contain* π. That would mean that $s_n(\pi)$, the number of permutations of length n *avoiding* π, is very close to $n!$. Since $(n+1)! = (n+1)n!$, you're multiplying by bigger and bigger numbers to get the next term in the factorial sequence; this means that $n!$ grows *faster* than exponential! Thus, the sequence $s_n(\pi)$ would not have (at most) exponential growth. However, it has been proven that this is never the case.

Theorem 2.9. (Stanley, Wilf) The *Stanley-Wilf conjecture* states that for every permutation π, there exists a number c such that $s_n(\pi) < c^n$ for all n. [8]

Remark 2.10. This is called the Stanley-Wilf *conjecture* because it was formulated independently by Richard P. Stanley and Herbert Wilf. Since then it was proven by Adam Marcus and Gábor Tardos and is no longer a conjecture, despite the name. For a proof, see (insert citation).

One thing that's of particular importance is the *smallest* c satisfying the above property. More precisely:

Definition 2.11. The *growth-rate* (or *Stanley-Wilf limit*) of a permutation π, denoted $L(\pi)$, is defined as

$$L(\pi) = \lim_{n\to\infty} \sqrt[n]{s_n(\pi)}.$$

Suppose we have some $l > L(\pi)$. Then, for sufficiently large n, $\sqrt[n]{s_n(\pi)} < l$. This is the same as saying $s_n(\pi) < l^n$ for sufficiently large n. By similar logic, if $l < L(\pi)$, then $s_n(\pi) > l^n$ for sufficiently large n. Thus $L(\pi)$ can be thought of as the number for which $s_n(\pi)$ is "roughly" equal to $L(\pi)^n$. It's important to note that even though the sequence $\sqrt[n]{s_n(\pi)}$ is *upper-bounded* by the Stanley-Wilf conjecture, this isn't enough to prove that it converges. For that, we need another theorem.

Theorem 2.12. For all permutations π, either

$$\lim_{n\to\infty} \sqrt[n]{s_n(\pi)} \in [1, \infty)$$

exists or it diverges to ∞. [1]

These two results combined show that $L(\pi)$ exists for all π.

Even though the growth rate of π always exists, in general it is *notoriously* difficult to find. For example, while the exact growth rates for 1234 and 1342 are known, the growth rate of 1324 remains an open problem. In 2012, it was proven that $L(1324) < 16$. [7] This upper bound has since been reduced to approximately 13.5 in 2017, along with a lower bound of about 10.271. [3]

3 Stack-Sortability of Permutations

Pattern avoidance has an interesting connection to a type of data structure used in computing. Take any permutation σ of length n. We want to find some algorithm that sorts the entries $\sigma_1, \ldots, \sigma_n$ of σ into the order $1, 2, \ldots, n$. (E.g. $2, 1, 4, 3$ gets sorted as $1, 2, 3, 4$.) There's a certain type of data structure we can use called a *stack*. In a stack, there are three sections: The input, the stack, and the output. At the beginning of the sorting process, we have the input permutation on the right. Next, there are two different manipulations we can preform on the stack.

1) We can move the leftmost number of the input string to the top of the stack. This operation is called

push.

2) We can move the number at the top of the stack to the right of the output string. This operation is called *pop.*

The following rules define an algorithm made of pushes and pops that attempts to sort the input permutation into increasing order:

1) The algorithm terminates when the stack and the input are empty.

2) Preform a push if the stack is empty and the input is not empty.

3) Preform a pop if the input is empty and the stack is not empty.

4) If x is at the top of the stack and y is on the left of the input, preform a pop if $x < y$ and a push if $x > y$.

Remark 3.1. Note that $x = y$ never happens since all elements of a permutation sequence are distinct.

Now, there are some permutations for which this algorithm works. An example is the permutation 3124. However, there are some that get sorted incorrectly. For example, the input 231 gets sorted into 213 rather than 123. This leads to the notion of *stack-sortability*:

Definition 3.2. A permutation σ on $[n]$ is said to be *stack-sortable* if the end result of the above algorithm is the permutation $123\cdots n$.

So, how do we know which permutations σ are stack-sortable? Well, the answer has a surprising connection to pattern avoidance!

Theorem 3.3. A permutation is stack-sortable if and only if it avoids the pattern 231. [2]

Proof. We will show the converse: A permutation is not stack-sortable if and only if it contains the pattern 231. Suppose we have a permutation σ on $[n]$ that contains the pattern 231. For any $a, b \in [n]$, write $a \prec b$ if a appears before b in the expression of σ; this can be written as

$$\sigma = \cdots a \cdots b \cdots ,$$

where $\cdots$ indicates that there are possibly some numbers in that region (but not necessarily). Now, what it means for σ to contain the pattern 231 is that there are some $i, j, k \in [n]$ such that $i < j < k$ and $j \prec k \prec i$. Let's choose such a triple $i, j, k \in [n]$ so that the distance between j and k in σ is as small as possible. This in particular means that there is no $a \in [n]$ such that $j \prec a \prec k$ and $k < a$. Then i, j, a would be a containment of 231 so that the distance between j and a along the string σ is less than that of j and k, a contradiction.

Now, by the time that k is on the far left of the input string, j is already either in the stack or the output string. Suppose it's still in the stack. Then, notice that any elements above j in the stack must have been pushed inside after j was. Thus they must appear to the right of j inside σ. But they must appear to the left of k inside σ since k hasn't been pushed in yet. Written mathematically, this means that for any a above j in the stack, $j \prec a \prec k$. By what we had earlier, this means that $a \leq k$. We can't have $a = k$ since $a \prec k$, so actually $a < k$.

What we've shown is that whenever k is on the far left of the input string, if j is in the stack then anything above j is $< k$. Thus the algorithm will preform a sequence of pop operations until j is popped. Thus there exists some instant that k is on the far left of the input string and j is in the output string. In that instant, any $a \succ k$ is also in the input string, including $a = i$. Thus in the end, i will appear to the right of j

in the output string, even though $i < j$. Thus the output string is not sorted, and σ is not stack-sortable.

Now for the other direction. Suppose that σ is not stack-sortable. Then, there exists a pair $i < j$ such that i appears to the right of j in the final output string. Now, let's rewind to the moment M_j right before j is popped off the stack. If i is in the output string at moment M_j, then j will appear to the right of i in the final output string, a contradiction. Thus that case can't happen. Suppose i is in the stack below j at M_j. In this case, let's look at moment M_i, the moment after i is pushed into the stack.

Because of M_j we know that j must've been pushed into the stack after i, so j is in the input string at moment M_i. After M_i, anything $< i$ on the left of the input string gets pushed in, until we arrive at the leftmost element $> i$ (which must exist since $j > i$ is in the input). Call this element a. Then, anything that's now on top of i gets popped off by a since it's $< a$. After that, i gets popped off. At this point, j is still in the input string since it's $> i$, so by the time M_j arrives, i is already in the output. This is a contradiction since we assumed i was below j in the stack at M_j. Thus that assumption leads to a contradiction.

We've now shown that at M_j, the element i can't be in the output string or in the stack below j. The only other possibility is that i is in the input string. Thus at moment M_j, there exists a leftmost element of the input string, call it k. Since j gets popped after M_j, we know that $j < k$. In particular, since $i < j$, this means that $i \neq k$. Thus k is before i in the input string at M_j. Since j is in the stack at M_j while k and i are in the input string, we know that j must've been to the left of k and i at the beginning of the algorithm. Thus we have $i < j < k$ and $j \prec k \prec i$, an appearance of the 231 pattern. □

4 Other Notions of Pattern Avoidance

Pattern avoidance is a concept that extends beyond permutations. Many other combinatorial structures have natural notions of "patterns" and "pattern avoidance". Here, we will list a few examples of other combinatorial structures, along with notions of pattern avoidance.

Suppose that n cars need to park in n parking spaces along a 1-way street. The cars are in a line along the street, with car 1 in the front, then car 2 behind it, etc. Each car has a favorite spot, and it will either take that spot or the next availible spot. A collection of prefrences that results in all the cars being parked is called a *parking function*.

Example 4.1. Suppose car 1 prefers spot 1, car 2 prefers spot 4, and car 3 and car 4 both prefer spot 2. This can be written as 1422, with the ith index representing the prefrence of car i. Now, 1422 is a parking function since all the cars get parked: car 1 in spot 1, car 2 in spot 4, car 3 in spot 2, and car 4 in spot 3. However, 2222 is not a parking function since the first three cars park in spots 2, 3, and 4, and car 1 has nowhere to park.

It's easy to see that if the word induced by a collection of prefrences is a permutation, then all the cars will have somewhere to park. Thus parking functions can be thought of as a generalization of permutations. Patterns in parking functions are defined similarly to patterns in permutations.

Definition 4.2. Let $\sigma = \sigma_1 \cdots \sigma_n$ and $\pi = \pi_1 \cdots \pi_m$ be parking functions. We say that π is a *pattern* of σ if there is a subsequence of the σ_k's that is order-isomorphic to π.

Definition 4.3. Let S be a set, called an *alphabet*, where the elements in S are called *symbols*. Then a *word* over S is a finite sequence of symbols from S. Let $\sigma = \sigma_1 \cdots \sigma_n$ and $\pi = \sigma_1 \cdots \sigma_m$ be two words over $\mathbb{N}$. Then π is a *pattern* of σ if there exists a subsequence of σ that's order-isomorphic to the sequence π.

Example 4.4. The word $\sigma = 12343$ is a valid word over $\mathbb{N}$. If $\pi = 122$, then the subsequence $2, 3, 3$ of σ is an instance of π. Thus σ contains π.

Now let's look at a structure that's not related to words.

Definition 4.5. A *graph* is a pair $G = (V, E)$ where V is a set of objects, called *vertices*, and $E \subseteq \mathcal{P}(V)$ is a set of two-element subsets of V, called *edges*. We say that $a, b \in V$ are *adjacent* if $\{a, b\} \in E$ (in particular, $a \neq b$). Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs. We say that G_2 is a *subgraph* of G_1 if there exists a subset $S \subseteq V_1$ and a bijection $f : S \to V_2$ such that $a, b \in S$ are adjacent if and only if $f(a), f(b)$ are adjacent. If G_2 is not a subgraph of G_1, then we say that G_1 *avoids* G_2.

Usually, when one talks about graphs $G = (V, E)$, it is assumed that V and E are finite sets. However, infinite graphs are sometimes studied. Now let's look at examples of different notions of pattern avoidance in permutations.

Definition 4.6. Let $\sigma = \sigma_1 \cdots \sigma_n$ and $\pi = \pi_1 \cdots \pi_m$ be permutations. If there is a sequence of consecutive σ's that is order-isomorphic to π, then π is said to be a *consecutive pattern* of σ.

Example 4.7. The permutation 132 is a consecutive pattern of 12534 because the consecutive sequence $2, 5, 3$ is order-isomorphic to $1, 3, 2$. Even though there is a subsequence $1, 2, 3$ in 12534, this does not consist of *consecutive* entries, so it is not an instance of 123 under this definition. However, $1, 2, 5$ is an instance of 123.

Definition 4.8. A *vincular pattern* is a permutation π containing dashes between some of the elements (e.g. $1 - 423 - 56$). We say that σ *contains* π as a *vincular pattern* if there is a subsequence of σ order-isomorphic to π such that adjacent entries in π without a dash between them must correspond to consecutive entries in σ.

Example 4.9. Let $\sigma = 13245$. Then the subsequence 124 is an instance of the pattern $\pi = 1 - 23$. However, 125 is not an instance of $1 - 23$, because the 2 and 5 aren't adjacent in σ. Even though all the entries are adjacent in 245, this is still an instance of $1 - 23$; the dash just means that the entries corresponding to 1 and 2 aren't *necessarily* adjacent, but they can be.

As you can see, there are many interesting notions of pattern avoidance outside the classical one.

Pattern avoidance is a rich subject, sometimes involving unexpected theorems and results. While lots of progress has been made in evaluating s_n functions and proving Wilf-eqivalences, there are many open problems yet to be solved. For example, the growth rate of 1324 is yet to be figured out exactly. All we can do is keep exploring open problems, proving new results, and continue learning new math.

Acknowledgements

I would like to thank my teacher Simon Rubinstein-Salzedo for making the writing of this paper possible. I would additionally like to thank my TA Alex Cao for giving me feedback to help make this paper better.

References

[1] Richard Arratia. On the stanley-wilf conjecture for the number of permutations avoiding a given pattern. *the electronic journal of combinatorics*, pages N1–N1, 1999.

[2] David Austin. Patterns in permutations. *American Mathematical Society*, 2017.

[3] David Bevan, Robert Brignall, Andrew Elvey Price, and Jay Pantone. A structural characterisation of av(1324) and new bounds on its growth rate, 2019.

[4] Miklós Bóna. Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *Journal of Combinatorial Theory, Series A*, 80(2):257–272, 1997.

[5] Mireille Bousquet-Mélou. Four classes of pattern-avoiding permutations under one roof: generating trees with two labels. *The electronic journal of combinatorics*, pages R19–R19, 2002.

[6] Alex Cao, Sparsho De, and Amit Saha. Pattern avoidance in permutations. *Euler Circle*, 2021.

[7] Anders Claesson, Vít Jelínek, and Einar Steingrímsson. Upper bounds for the stanley–wilf limit of 1324 and other layered patterns. *Journal of Combinatorial Theory, Series A*, 119(8):1680–1691, November 2012.

[8] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the stanley–wilf conjecture. *Journal of Combinatorial Theory, Series A*, 107(1):153–160, 2004.

[9] Brian Koichi Nakamura. *Computational methods in permutation patterns*. Rutgers The State University of New Jersey, School of Graduate Studies, 2013.

[10] Amitai Regev. Asymptotic values for degrees associated with strips of young diagrams. *Advances in Mathematics*, 41(2):115–136, 1981.