

18.212: Algebraic Combinatorics

Lecturer: Professor Alexander Postnikov

Notes by: Andrew Lin

Spring 2019

Introduction

The website for this class can be found [here](#) (**note**: link now redirects to the set of all past semesters) – we should check it from time to time, because there will be some links to additional readings. Problem sets will also be posted there. There will be no exams in this class; 18.212 is graded entirely on four or five problem sets (assigned every two or three weeks). Overall, there are two conflicting goals for this class: everyone should understand everything, so we shouldn't go too fast, but the class shouldn't be too slow and boring. (So we should let Professor Postnikov know if we feel one way or the other!)

As the title suggests, this is a class on combinatorics. **Combinatorics** is the area of mathematics that studies discrete objects, like graphs, permutations, and various diagrams, looking at objects that we can count or list. These days, there are two main flavors: Stanley-style and Erdős-style. **Stanley-style** (also **enumerative, algebraic, or geometric**) combinatorics deals with counting objects or their connections with algebra and geometry, and that's what we'll see in this class. On the other hand, there is also **Erdős-style** (also **extremal** or **probabilistic**) combinatorics, which is explored in a class like 18.218.

Fact 1

In our class, we'll often show results of the form $A = B$, saying that some number of objects is equal to some other number. For example, **Cayley's formula** states that the number of labeled trees on n vertices is n^{n-2} , and the proof of this result gets us to that exact value. On the other hand, Erdős-style combinatorics cares about things like "what is the probability a random graph is connected?", which involves stating and proving statements like $A \sim n \log n$ instead.

No textbooks are required for this class. The recommended textbooks for this class are the following (which can be located on the class website):

- Richard Stanley's *Algebraic Combinatorics*,
- Richard Stanley's *Enumerative Combinatorics*, volumes 1 and 2 (graduate-level book with lots of material),
- van Lint and Wilson's *A Course in Combinatorics* (a very nice book but a bit outdated).

Remark 2. *These notes were edited during December 2021 for clarity and corrections; apologies for any additional typos that have come up.*

1 February 6, 2019

We're going to start by studying **Catalan numbers**, which are not actually part of Stanley's *Algebraic Combinatorics*. On the other hand, Professor Stanley did compile hundreds of Catalan number interpretations separately from his book, so he "has an excuse."

Definition 3

The n th **Catalan number** C_n is the number of sequences $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{2n})$, where all $\varepsilon_i \in \{-1, 1\}$, there are exactly n 1s and n -1s, and the partial sums $\varepsilon_1 + \dots + \varepsilon_i$ are nonnegative for all $1 \leq i \leq 2n$.

Such sequences are often represented by **Dyck paths**, which contain **up** and **down** edges. Each up step corresponds to a vector $(1, 1)$, and each down step corresponds to $(1, -1)$. Lining up all of the vectors, we have a path from $(0, 0)$ to $(2n, 0)$, where the entire path is contained in the upper-half plane (including the x -axis).

It is easy to count the number of paths directly for small n with casework, and we get the following numbers:

n	0	1	2	3	4	5
C_n	1	1	2	5	14	42

There is an explicit formula that we will get to, but we'll give an example to illustrate the use of objects like this. This is getting at the question of "what is mathematics" – if objects have no real-world applications, they might be meaningless, so here's a "real-world" application of Catalan numbers.

Example 4 (Drunk man problem)

A drunk man is on a (one-dimensional) road, and he cannot tell left from right, so he has a $\frac{1}{2}$ chance of going to the left by 1 and a $\frac{1}{2}$ chance of going to the right by 1 at each step. (This is the simplest example of a **random walk**.) The man starts at $x = 1$, and there is a cliff starting at $x = 0$. What is the probability he survives forever?

One way to compute this is to instead calculate the probability that the man falls – he can immediately take one step to the left, or one step to the right and two steps to the left, and so on. And the key observation is that **these kind of paths correspond exactly to Dyck paths**. Specifically, if the drunk guy falls, he takes $2n$ steps (for some n) and then has a $\frac{1}{2}$ chance of taking that last $2n + 1$ th step, and those first $2n$ steps must stay within the region $x > 0$. So the probability that we're looking for is

$$P = \frac{1}{2} \sum_{n \geq 0} \frac{C_n}{2^{2n}}.$$

As a slight generalization, we can assume the road is tilted at some slope, so that the drunk man goes left and right with different probabilities $1 - p$ and p , respectively. This gives instead

$$P = p \sum_{n \geq 0} C_n p^n (1 - p)^n,$$

and the setting is called a **biased random walk** – we are essentially interested in seeing whether it's possible to evaluate this explicitly.

But before we dive into calculations, we should note that the solution to this problem doesn't require Catalan numbers. Here's an elementary method:

Solution. As before, let's say the cliff is at $x = 0$, and the man starts at $x = 1$. Let's imagine that there is a house where the drunk man lives at $x = N$, and he is safe if he reaches it before falling. Let $P(i, N)$ denote the probability

that the man survives if he starts at position i and his house is at position N . (Our goal is actually to calculate the probability when $N \rightarrow \infty$.)

First of all, notice that $P(N, N) = 1$ and $P(0, N) = 0$. We also have the relation (for all $0 < i < N$)

$$P_i = \frac{1}{2}(P_{i-1} + P_{i+1}),$$

since from position i , there's a one-half chance of going to position $i - 1$ and a one-half chance of going to position $i + 1$. So we actually have an arithmetic progression, and $P(i, N) = \frac{i}{N}$. But in our initial problem, there is no house, so we can take the limit as $N \rightarrow \infty$, and this yields $P(1, N) \rightarrow 0$. Thus, the man will always fall off the cliff with probability 1. \square

(One potential moral of this story is “don’t drink if you are at the edge of a cliff, especially if you’re under 21 years old.”) But this proof isn’t so easy to generalize to the biased walk, and we can also learn more about the structure of the generalized problem if we use Catalan numbers. So let’s start by finding a way to express each Catalan number in terms of the previous ones:

Proposition 5 (Recurrence relation for the Catalan numbers)

We have the base case $C_0 = 1$, and for all $n \geq 1$,

$$C_n = \sum_{k=1}^n C_{k-1}C_{n-k}.$$

Proof. The base case is clear. For the inductive step, we want to somehow break a Dyck path into smaller Dyck paths. Dyck paths are allowed to touch (but not cross) the x -axis before $(2n, 0)$, so we find the first place after $x = 0$ where the path touches the x -axis and call it $(2k, 0)$. (It will be an even x -coordinate, because we need an equal number of up and down moves.)

The part of the path from 0 to $2k$ is strictly above the x -axis, so it is an up move, a Dyck path on $2k - 2$ moves (C_{k-1}), and then a down move, followed by a Dyck path on $2n - 2k$ moves (multiply by C_{n-k}). Furthermore, this decomposition of a given Dyck path is unique, because we specified that we took the first point of contact with the x -axis. Since k can be anything from 1 to n inclusive, this gives our recurrence relation as desired. \square

We can actually write this relation as a **generating function**: given any sequence (c_0, c_1, \dots) , we can construct a generating function

$$f(x) = c_0 + c_1x + c_2x^2 + \dots.$$

We can ask questions of convergence or divergence, but in combinatorics we avoid these questions by treating these as **formal power series**, which are essentially infinite sequences of coefficients, so that we can add and multiply the series in a well-defined way.

Proposition 6 (Generating function for the Catalan numbers)

Let $f(x)$ be the generating function for the Catalan numbers. The recurrence relation in Proposition 5 is equivalent to

$$f(x) = xf(x)^2 - 1.$$

Proof. We’ll extract coefficients on the left and right side and see that they are the same: on one side, we have

$$[x^n]f(x) = C_n$$

(where this notation means the x^n coefficient of f). For the other side, first note that

$$[x^n]xf(x) = C_{n-1},$$

since we're shifting our coefficients $C_{k-1}x^{k-1} \rightarrow C_{k-1}x^k$. This means that

$$[x^n]f(x)^2 = C_n \cdot C_0 + C_{n-1} \cdot C_1 + \cdots + C_0 \cdot C_n$$

by expanding out our coefficients, and

$$[x^n]xf(x)^2 = C_{n-1} \cdot C_0 + C_{n-2} \cdot C_1 + \cdots + C_0 \cdot C_{n-1} = C_n$$

by our recurrence relation. So all coefficients $n \geq 1$ match up, and finally we just check our constant term (which is 1 on both sides). Since all coefficients are the same, the power series are indeed equal. \square

And now we can solve our functional equation $f(x) = xf(x)^2 + 1$ by treating it as a quadratic in f . We find, by the quadratic formula, that

$$xf^2 - f + 1 = 0 \implies f = \frac{1 \pm \sqrt{1 - 4x}}{2x}.$$

There may appear to be some ambiguity about the \pm sign, but $f(x)$ is a power series with constant term 1, so $f(x)$ can't diverge. Thus, we need to make sure the numerator is equal to 0 at $x = 0$ as well, and we have our final generating function

$$f(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

We'll explore this more next time!

2 February 8, 2019

Last time, we defined the Catalan numbers and bijected them to Dyck paths. We discovered a recurrence relation for these Catalan numbers, and from that, we found our generating function

$$f(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

Let's see an application of this generating function to the biased random walk problem (again, in which we are at the very edge of a cliff, and we have a probability p of walking to the right and $1 - p$ of walking to the left).

Proposition 7

The probability that the man falls down the cliff in the biased random walk is

$$(1 - p)f(p(1 - p)) = \begin{cases} 1 & p \leq \frac{1}{2}, \\ \frac{1-p}{p} & p > \frac{1}{2}. \end{cases}$$

(In other words, "if we're going to drink at the edge of a cliff, make sure it slopes down to the right.") We'll start with a fact that many of us probably know well:

Proposition 8 (Binomial theorem)

For any real number y and nonnegative integer a , we have

$$(1 + y)^a = 1 + \binom{a}{1}y + \binom{a}{2}y^2 + \binom{a}{3}y^3 + \dots$$

(This can be proved by Pascal's identity and induction.) In fact, the binomial theorem is actually true for **any** value of a (not just integers), so Pascal's triangle is part of the picture, but it doesn't tell us the whole story. We just need to redefine (for a nonnegative integer k) the **binomial coefficient**

$$\binom{a}{k} \equiv \frac{a(a-1)(a-2)\cdots(a-k+1)}{k!}.$$

The binomial theorem then follows by writing out a Taylor expansion of the left hand side: the $a(a-1)(a-2)\cdots$ comes from the power rule. So we can apply this to our generating function by setting $y = -4x$, $a = \frac{1}{2}$: we ignore the constant term, and for all $n \geq 1$,

$$f(x) = \frac{1 - (1 - 4x)^{-1/2}}{2x} \implies [x^n]f(x) = -\frac{1}{2}[x^{n+1}](1 - 4x)^{1/2},$$

which is (by the binomial theorem)

$$[x^n]f(x) = -\frac{1}{2} \binom{1/2}{n+1} (-4)^{n+1}.$$

After expanding out the binomial coefficient, this finally gives us a nice formula:

Theorem 9 (Closed form for the Catalan numbers)

The n th Catalan number is

$$C_n = \frac{\binom{2n}{n}}{n+1}.$$

But this is a very nice expression, so we may want to ask if there is a combinatorial proof instead.

Proof by reflection principle. Recall that C_n counts the number of Dyck paths that are above the x -axis from $(0, 0)$ to $(2n, 0)$. First of all, if we remove the "above the x -axis" condition, we just want the number of paths that have n ups and n downs: this is just $\binom{2n}{n}$. We now want to remove all paths that were overcounted, which is the same as counting the number of non-Dyck paths from $(0, 0)$ to $(2n, 0)$. Each of these paths must intersect the line $y = -1$ at least once – **pick the first intersection**, and take the portion of the path from that intersection to the end $(2n, 0)$. We then reflect that portion over $y = -1$, and now the endpoint of our path is $(2n, -2)$.

This new path has $n - 1$ up steps and $n + 1$ down steps, and set of all such paths (up-down paths from $(0, 0)$ to $(2n, -2)$) are in bijection with the non-Dyck paths! This is because paths from $(0, 0)$ to $(2n, -2)$ always cross $y = -1$, so we can find the first intersection and do the reflection again to get to our non-Dyck path from $(0, 0)$ to $(2n, 0)$. Since there are $\binom{2n}{n-1}$ such paths, the number of Dyck paths is

$$C_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1} \binom{2n}{n}.$$

□

However, as a combinatorialist, Professor Postnikov doesn't like subtraction, so here's a proof that doesn't even require that:

Proof by necklaces. Start with the binomial coefficient $\binom{2n+1}{n}$, which counts the number of sequences $(\epsilon_1, \dots, \epsilon_{2n+1})$ with $n+1$ 1s and $n+1$ -1s.

For illustration, consider the sequence $(+, -, -, +, -)$ for $n = 2$. For each of the $\binom{2n+1}{n}$ sequences, consider all $(2n+1)$ different **cyclic shifts** of it (that is, moving the first term of the sequence to the end repeatedly): we get $(-, -, +, -, +)$, $(-, +, -, +, -)$, $(+, -, +, -, -)$, and $(-, +, -, -, +)$ in addition to the original sequence. For another perspective, think about arranging +s and -s on a necklace (as black and white beads, respectively). Then the necklace has n black beads and $n+1$ white beads, and each of the cyclic shifts corresponds to cutting the necklace a different spot.

Claim 10. *All cyclic shifts are different.*

The idea is that if two different cyclic shifts were the same, we'd be able to decompose the necklace into multiple copies of a subsequence, which is bad because the difference in the number of +s and -s is 1. But the details are left as an exercise for us, and this means the number of total **necklaces** that exists is $\frac{\binom{2n+1}{n}}{2n+1}$.

Claim 11. *For each necklace, there is a unique sequence $(\epsilon_1, \epsilon_2, \dots, \epsilon_{2n+1})$ among all $2n+1$ ways to cut the necklace, such that $\epsilon_1, \dots, \epsilon_{2n}$ form a Catalan sequence and $\epsilon_{2n+1} = -1$.*

This is also left as an exercise, but the idea is to find the "lowest point" on the Dyck path. And once both of the claims above are resolved, notice that the number of Catalan sequences is therefore also equal to

$$\frac{\binom{2n+1}{n}}{2n+1} = \frac{1}{n+1} \binom{2n}{n},$$

as desired. □

This is a "good" combinatorial proof, because we're actually doing some kind of natural division, where the cyclic shifts explain how the binomial coefficient is reduced. And now we'll get to see how the Catalan numbers pop up basically everywhere:

Fact 12

C_n is the number of triangulations of an $(n+2)$ -gon.

For example, given a hexagon, there are $C_4 = 14$ ways to cut it into 4 triangles. Indeed, there are 6 of them where all three diagonals drawn come from one vertex, 6 where they form a zig-zag pattern, and 2 by drawing an equilateral triangle.

Fact 13

C_n is the number of valid parenthesizations of $n+1$ letters.

For illustration, take $n = 4$: the idea is that we want to put some parentheses in the expression

$$a b c d e.$$

One such way is $((a(bc))(de))$,

Fact 14

C_n is the number of plane binary trees with n non-leaf vertices and $n+1$ leaves.

Basically, we start with the root of a tree, and repeatedly take an existing leaf and draw a left and right child for it. We can show that the number of leaves and non-leaves will always differ by 1, and that this is the only way to create such plane binary trees.

Proposition 15

There is a bijection between the three countings of C_n above.

Proof. Starting with an $(n + 2)$ -gon, the binary tree will be the **dual graph** to the triangulation: pick one of the sides of our polygon to start constructing our tree. Non-leaf vertices will correspond to triangles, and the triangle starting to the edge is our root. This means we can convert any triangulation to a tree and vice versa. Then we can take our binary tree and use “common neighbor” as a set of parentheses, where we take leaves from left to right, and this completes the bijection between trees and parenthesizations. \square

The bijection from these three constructions to Dyck paths is less obvious, so that will be an exercise for us.

3 February 11, 2019

Remark 16. *As MIT students, many of us may know a lot about computer science. The Bible of computer science is “The Art of Computer Programming” by Knuth, and a lot of the material of this class comes from it.*

We'll continue to see connections to Catalan numbers today through other mathematical objects:

Definition 17

A **queue** is a data structure that is “first-in-first-out,” while a **stack** is “last-in-first out.”

In words, a queue contains one or several entries, and it's like a line: if a person enters the line first, it will exit first. Meanwhile, a stack is like a pile of papers, so the paper that was most recently added to the pile will be removed first. It turns out that we can use these data structures to sort permutations, and Catalan numbers will appear again.

Definition 18

A permutation is **queue-sortable** (resp. **stack-sortable**) if we can read it from beginning to end, either putting numbers in the queue (resp. stack) or directly into a final list, and also popping elements out of the queue (resp. stack) whenever we'd like, and ending up with a sorted list.

Proposition 19

The number of **queue-sortable** permutations of $(1, 2, \dots, n)$ is equal to C_n .

For example, for $n = 4$, $(2, 4, 1, 3)$ is queue-sortable with the following procedure: put 2, then 4 in the queue, then put 1 in our list directly. After that, take 2 out of the queue, put 3 in the list directly, and finally take 4 out of the queue. On the other hand, an example of something not queue-sortable is $(3, 2, 1)$, since we would have to put 3 in the queue and then 2, and that's bad because 3 must come out first.²

Proposition 20

The number of **stack-sortable** permutations of $(1, 2, \dots, n)$ is also equal to C_n .

For example, $(4, 1, 3, 2)$ is stack-sortable, since we put 4 in the stack, put 1 in the list directly, then put 3 and 2 in the stack and pop everything back out. But $(2, 3, 1)$ is not stack-sortable.

In the above situation, $(3, 2, 1)$ and $(2, 3, 1)$ are essentially fundamental “patterns” of the numbers, and the idea of sortability is related to the more general concept in combinatorics of **pattern avoidance**:

Definition 21

Given a permutation $w = (w_1, w_2, \dots, w_n)$ of size n (so w is an element of the **symmetric group** S_n , which we'll discuss later in the class), and given a permutation $\pi = (\pi_1, \pi_2, \dots, \pi_k)$ of size $k \leq n$, we say that w **contains** pattern π if there exists a (not-necessarily-consecutive) **subsequence** of entries $(w_{i_1}, w_{i_2}, \dots, w_{i_k})$ whose values are in the same relative order as π . Meanwhile, w is **π -avoiding** if it does not contain the pattern π .

Example 22

The permutation $w = (3, 5, 2, 4, 1, 6)$ does contain the pattern $\pi = (2, 1, 3)$ (consider indices 2, 3, 6 with values 5, 2, 6, for example), but it is $(1, 2, 3, 4)$ -avoiding (there is no subsequence of four increasing values).

Proposition 23

Queue-sortable permutations are exactly those permutations that are 321-avoiding, and stack-sortable permutations are those that are 231-avoiding. In fact, for any pattern π of size 3, the number of π -avoiding permutations is C_n .

This is once again left as an exercise for us!

Remark 24. Note that we will not be required to solve all problems in a problem set, so some will be easier and some will be harder. The goal is for there to be something interesting for all of us.

With this, it's time to move to the next topic – we're going to talk about **partitions, Young diagrams, and Young tableaux**.

Remark 25. Since the word “tableau” is French, we add an “x” to the end to make it plural.

Definition 26

A **partition** of n is a list of integers $\lambda = (\lambda_1, \dots, \lambda_e)$, such that $n = \lambda_1 + \dots + \lambda_e$, the λ_i s are weakly decreasing (meaning $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_e$), and all λ_i are positive integers.

Because we're discussing **partitions** of n , rather than **compositions** of n , the order of elements in our list doesn't matter. So it's just by convention that people write them this way from largest to smallest.

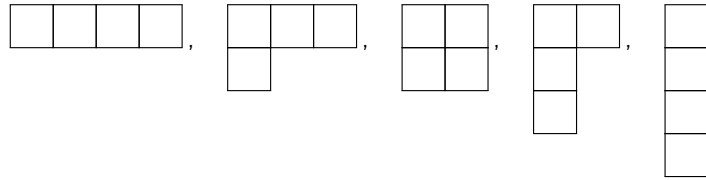
Example 27

There are 5 partitions of 4, corresponding to writing $4 = 1 + 1 + 1 + 1 = 2 + 1 + 1 = 2 + 2 = 3 + 1 = 4$.

Definition 28


A **Young diagram** is a shape corresponding to the partition $(\lambda_1, \dots, \lambda_k)$, where there are k left-justified rows of $\lambda_1, \lambda_2, \dots, \lambda_k$ boxes, respectively. A similar term, **Ferrers shapes**, refers to similar diagrams with dots instead of boxes.

Here's what the Young diagrams look like for the partitions of 4:



Definition 29

A **standard Young tableau** (SYT) is a way to fill in the n boxes of a Young diagram with the numbers $1, 2, \dots, n$ (without repetition), such that the numbers are increasing across rows and down the columns.

For example, if $\lambda = (4, 2, 2, 1)$, our tableau looks like . By abuse of notation, we'll use λ for its corresponding Young diagram as well, and here's an example of a standard Young tableau for that partition:

1	3	4	7
2	6		
5	9		
8			

Definition 30

Let f^λ denote the number of standard Young tableaux of shape λ .

Lemma 31

For $\lambda = (n, n)$, the number of standard Young tableaux is $f^{(n,n)} = C_n$.

Proof. Given a standard Young tableau, construct a corresponding sequence $(\epsilon_1, \dots, \epsilon_{2n})$ via

$$\epsilon_i = \begin{cases} + & \text{if } i \text{ is in the first row} \\ - & \text{if it is in the second row} \end{cases}$$

These are exactly the sequences that correspond to Dyck paths! For example, the following Young tableau corresponds to the Dyck path $(+, +, -, +, -, -, +, +, -, -)$:

1	2	4	7	8
3	5	6	9	10

Any Young tableau will correspond to a Dyck path, since we always have at least as many integers in the top row as the bottom up to some $k \leq 2n$. And any Dyck path will correspond to a Young tableau (by adding numbers in the top or bottom row, respectively). Thus the number of standard Young tableaux of shape (n, n) is indeed the number of Dyck paths, which is C_n . □

This formula therefore means that we can think of counting Young tableaux as giving extensions of the Catalan numbers. And in fact, there turns out to be a nice formula for general λ

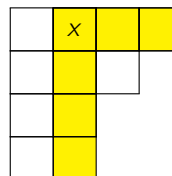
Theorem 32 (Hook length formula (Frame–Robinson–Thrall))

The number of standard Young tableaux for a partition λ is

$$f^\lambda = \frac{n!}{\prod_{x \in \lambda} h(x)},$$

where $h(x)$ are the **hook lengths** of the squares x ; that is, the number of squares in a hook that goes to the right and down from square x . The **arm** is the length of the horizontal component not including x , and the **leg** is the length of the vertical component not including x .

For example, the hook below has length 6, arm length 2, and leg length 3:

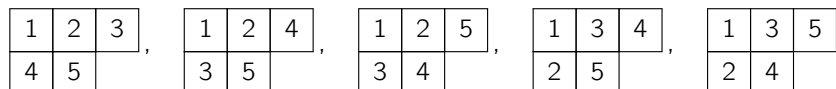


Example 33

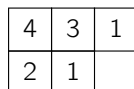
Take the partition $\lambda = (3, 2)$, so the Young diagram looks like

.

The possible Young tableaux for this shape are



Meanwhile, towards applying the hook-length formula, here are the hook lengths for each square in the tableau:



The number of possible standard Young tableaux is indeed $f^\lambda = \frac{5!}{4 \cdot 3 \cdot 1 \cdot 2 \cdot 1} = 5$, as expected. There are many proofs of the hook length formula, and the result initially comes from representation theory. However, there is a recent probabilistic proof using random walks, which we will cover later in this class.

4 February 13, 2019

Remark 34. *A few students asked about the problem set – it will be assigned sometime in the near future, and it'll probably due around the end of February or beginning of March.*

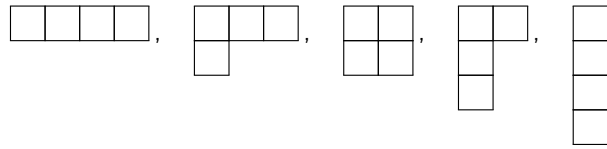
Remember that we defined f^λ to be the number of standard Young tableaux of shape λ . It turns out that these numbers are related to the symmetric group, but we'll discuss that later. The magic is that f^λ has a closed form, given by the hook length formula; as it turns out, exploring this path will lead us to some other cool results too.

Proposition 35 (Frobenius-Young identity)

For any integer n , we have

$$\sum_{\lambda:|\lambda|=n} (f^\lambda)^2 = n!.$$

This result also comes from representation theory, though we won't need that for the proof that we present. As an example, if we take $n = 4$, the possible Young diagrams are



The number of ways to fill in those Young diagrams with Young tableaux are 1, 3, 2, 3, 1 respectively, and indeed

$$1^2 + 3^2 + 2^2 + 3^2 + 1^2 = 24 = 4!.$$

To work towards a proof, we must first interpret this identity – the right hand side of the equation is the number of permutations in S_n (on n elements). Meanwhile, the left hand side counts **pairs of Young tableaux** p, q , such that p and q are both SYTs of the same shape λ . So it's natural to ask whether there's a bijection that we can set up between the two sets.

The tool we'll use is the **Schensted correspondence** from 1961, which was generalized later by Knuth to the **Robinson-Schensted-Knuth** correspondence (which is about semi-standard Young tableaux)! This is a pretty central construction in algebraic combinatorics, and for more details, we can see the papers on the course webpage.

Proof. Given a standard Young tableau T filled with some positive integers in S , suppose we want to add another positive integer x . The **Schensted's insertion algorithm** goes as follows: let $T \leftarrow x$ be the standard Young tableau with an extra box added to T , given by the following procedure:

1. Initialize $x_1 = x$ and $i = 1$.
2. If x_i is greater than all entries in the i th row of T , then add a new box at the end of the i th row containing x_i and stop.
3. Otherwise, find the smallest entry y in the i th row that is greater than x_i . Replace y by x_i , increment i by one, and go back to step 2.

The idea is that when we insert x , we may bump the entry x_2 down to the next row. Then we insert x_2 , which may bump something else, and we keep doing this until no bump occurs.

With this algorithm, given some permutation $w \in S_n$, let's construct a **pair** (P, Q) of Young tableaux. P is called the **insertion tableau**, and Q is the **recording tableau**.

To construct P , start with the empty tableau and add w_1, w_2, \dots, w_n one at a time, in that order, using the algorithm above. At the same time, we construct Q (of the same shape as P) by having i placed in the box that was added at the i th step of constructing P (that is, we record which box is added).

Example 36

Take the permutation $w = (3, 5, 2, 4, 7, 1, 6)$; we'll work through the corresponding (P, Q) that is created.

After we insert 3 and 5, we get

$$P_0 = \emptyset \implies P_1 = \boxed{3} \implies P_2 = \boxed{3 \ 5}.$$

Trying to insert 2 kicks 3 out into the next row:

$$P_3 = \begin{array}{|c|c|} \hline 2 & 5 \\ \hline 3 & \\ \hline \end{array}$$

Now 4 bumps 5 to get

$$P_4 = \begin{array}{|c|c|} \hline 2 & 4 \\ \hline 3 & 5 \\ \hline \end{array}$$

and then adding 7, 1, 6 in that order give us

$$P_5 = \begin{array}{|c|c|c|} \hline 2 & 4 & 7 \\ \hline 3 & 5 & \\ \hline \end{array} \implies P_6 = \begin{array}{|c|c|c|} \hline 1 & 4 & 7 \\ \hline 2 & 5 & \\ \hline 3 & & \\ \hline \end{array} \implies P_7 = P = \begin{array}{|c|c|c|} \hline 1 & 4 & 6 \\ \hline 2 & 5 & 7 \\ \hline 3 & & \\ \hline \end{array}$$

To find the corresponding Q -tableau, we scan over the process and see when each box was added:

$$Q = \begin{array}{|c|c|c|} \hline 1 & 2 & 5 \\ \hline 3 & 4 & 7 \\ \hline 6 & & \\ \hline \end{array}$$

□

Theorem 37

The construction $w \mapsto (P, Q)$ is a bijection between the symmetric group S_n and the set of all pairs (P, Q) of Young tableaux with identical shapes and n boxes.

Proof by illustration. To show that $w \mapsto (P, Q)$ is a bijection, we want to show that given a pair of tableau P and Q , we can reconstruct the original permutation. Here's the reverse procedure: start with, for example,

$$P = \begin{array}{|c|c|c|} \hline 1 & 4 & 6 \\ \hline 2 & 5 & 7 \\ \hline 3 & & \\ \hline \end{array}, \quad Q = \begin{array}{|c|c|c|} \hline 1 & 2 & 5 \\ \hline 3 & 4 & 7 \\ \hline 6 & & \\ \hline \end{array}$$

The position of the 7 in Q tells us that that box was added last in our algorithm. Because this box is in the second row, we must have had exactly one “bump” take place, and that bump displaced 7. And the entry that bumped 7 must have been 6, the maximal entry of the previous row smaller than 7. In other words, 6 was just added at the end of the process, and that means w ended with 6. Removing its contribution to P and Q gives us the intermediate step

$$P' = \begin{array}{|c|c|c|} \hline 1 & 4 & 7 \\ \hline 2 & 5 & \\ \hline 3 & & \\ \hline \end{array}, \quad Q' = \begin{array}{|c|c|c|} \hline 1 & 2 & 5 \\ \hline 3 & 4 & \\ \hline 6 & & \\ \hline \end{array}$$

Now keep repeating this logic: the bottom left corner has the next maximal entry of Q , so 3 in the third row was bumped in the previous step. Looking above, it must have been bumped by 2, and that must have been bumped from 1, so w 's second-to-last entry is 1. Continuing this procedure, we can recover w by peeling off the sequence from the end! So given any pair of tableaux, we can uniquely reconstruct w , showing the bijection. □

This correspondence has other special applications as well – one such application is for studying increasing and decreasing subsequences of permutations.

Definition 38

Given a permutation w that corresponds to a pair (P, Q) under the Schensted correspondence, the shape λ of P (and Q) is called the **Schensted shape** of w .

The first entry λ_1 , which is the number of boxes in λ 's first row, has a special significance:

Theorem 39 (Schensted)

The length of the first row of λ is the length of the longest increasing subsequence in w . Also, as a dual statement, the length of the first column of λ is the length of the longest decreasing subsequence in w .

For example, the first row has 3 boxes in the example above, so the longest increasing subsequence has length 3. Similarly, the first column has 3 boxes, so the longest decreasing subsequence also has length 3.

We'll wait until next lecture to do this proof. Instead, we'll show evidence of a special case Proposition 23 with another proof:

Corollary 40

The number of 123-avoiding permutations in S_n is the Catalan number C_n .

Proof. A permutation w is 123-avoiding if and only if the length of the longest increasing subsequence is at most 2. Thus, the Schensted correspondence sends any such permutation to a pair of tableaux with at most 2 columns, where both P and Q have n boxes. We want to biject this to a single tableau of shape $(2, 2, \dots, 2)$, since we already know that there are C_n ways to construct Young tableaux of that shape by Lemma 31 (since the shape (n, n) and $(2, 2, \dots, 2)$ can be filled in identical ways by reflection).

To construct that bijection, rotate Q by 180 degrees and stick it in place with P , except replacing any entry k with $(2n + 1) - k$. For example,

$$\begin{array}{|c|c|} \hline 1 & 3 \\ \hline 2 & 4 \\ \hline 5 & \\ \hline \end{array} + \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 5 \\ \hline 4 & \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 3 \\ \hline 2 & 4 \\ \hline 5 & 7 \\ \hline 6 & 8 \\ \hline 9 & 10 \\ \hline \end{array}$$

We may check ourselves that this is a valid construction (and thus the number of 123-avoiding permutations is indeed the number of Dyck paths C_n). □

Finally, here's a generalization of the Schensted theorem. We have discussed an interpretation of λ_1 , but we can figure out the whole shape of λ by looking at just certain properties of w :

Theorem 41 (Greene)

For any permutation and corresponding Schensted shape λ , $\lambda_1 + \lambda_2$ is the maximal subsequence that can be covered by two increasing subsequences, $\lambda_1 + \lambda_2 + \lambda_3$ is the maximal subsequence covered by three subsequences, and so on. Furthermore, the same statement also holds for columns and decreasing subsequences.

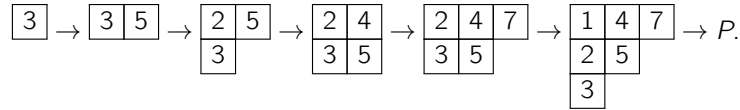
(If we're curious about this, we can read more by searching things up in the literature!)

5 February 15, 2019

Recall the Schensted correspondence between permutations $w \in S_n$ and pairs of Young tableaux (P, Q) : for example, we found that $w = (3, 5, 2, 4, 7, 1, 6)$ corresponds to

$$P = \begin{array}{|c|c|c|} \hline 1 & 4 & 6 \\ \hline 2 & 5 & 7 \\ \hline 3 & & \\ \hline \end{array}, Q = \begin{array}{|c|c|c|} \hline 1 & 2 & 5 \\ \hline 3 & 4 & 7 \\ \hline 6 & & \\ \hline \end{array}$$

through the insertion process



As mentioned, according to Schensted's theorem, the shape $\lambda = (3, 3, 1)$ tells how many boxes are in each row, and λ_1 , the number of boxes in the first row, is the size of the longest increasing subsequence in w . Similarly, the shape $\lambda' = (3, 2, 2)$ (the "transpose" of λ) tells us how many boxes are in each column, and λ'_1 , the number of boxes in the first column, is the size of the longest decreasing subsequence in w .

We're going to prove the first half of this (increasing subsequence) and leave the other half as an exercise!

Definition 42

The **j th basic subsequence** B_j in a permutation w (for any $1 \leq j \leq \lambda_1$) consisting of all entries of w , in order, that were inserted **initially** into the j th column during the insertion algorithm. In other words, B_j contains the set of entries that occupied the j th box in the first row after some insertion $T \leftarrow x$.

For example, we have 3 basic subsequences in the example above: for B_1 , we inserted 3, 2, and 1, so $B_1 = (3, 2, 1)$. Similarly, $B_2 = (5, 4)$, and $B_3 = (7, 6)$. (Note that the permutations do not necessarily need to cover disjoint intervals of integers – that's just what happened in this particular case.)

Lemma 43

Each B_j is a decreasing sequence.

Proof. The basic subsequence B_j contains the numbers that occupy the j th box in the first row, so to be added to the j th basic subsequence after the first element of it, we must bump the number in that spot. But bumping can only occur if we have a smaller number, so B_j is indeed decreasing. □

Lemma 44

For all $j \geq 2$, given any $x \in B_j$, we can find $y \in B_{j-1}$ such that $y < x$ and y is located to the left of x in the permutation w .

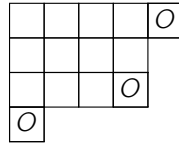
Proof. Consider the moment of insertion of x , and take y to be the entry that was located to the left of x . It must be less than x (by the rules of our algorithm), and it appears before x in the permutation because it was already inserted. □

Combining these two facts gives us the desired result:

Proof of Theorem 39, part 1. λ_1 is the number of basic subsequences by definition. Note that for any increasing subsequence $x_1 < x_2 < \dots < x_r$, we can only have at most 1 entry from each B_j (because the B_j s are decreasing by Lemma 43), so we must have $r \leq \lambda_1$. It suffices to show that an increasing subsequence of length λ_1 exists. For this, pick the last basic subsequence B_{λ_1} , and pick any $x_{\lambda_1} \in B_{\lambda_1}$. By Lemma 44, there is some $x_{\lambda_1-1} \in B_{\lambda_1-1}$ that comes before x_{λ_1} , and thus some $x_{\lambda_1-2} \in B_{\lambda_1-2}$ that comes before x_{λ_1-1} , and so on. Repeating this process gives us an increasing subsequence of length λ_1 , as desired. □

We'll take the remaining time in this lecture to prove the hook length formula (Theorem 32). The original proof was pretty complicated and relied on some other formulas. But later on, simpler proofs were found – we'll discuss one that uses a random walk.

Hook walk proof by Greene–Nijenhuis–Wilf (1979). There is a recurrence relation for the number of standard Young tableaux. Given a tableau with n boxes, n must appear in one of the bottom-right corner boxes, and removing this corner gives us a standard Young tableau of size $n - 1$. For instance, consider the following Young diagram for $\lambda = (5, 4, 4, 1)$:



Since the O 's are the spots that the largest number can be in, we have the recurrence relation

$$f^{5441} = f^{4441} + f^{5431} + f^{544}.$$

Thus, to prove the hook length formula, **it suffices to establish the recurrence relation**

$$f^\lambda = \sum_{\nu \text{ corner of } \lambda} f^{\lambda-\nu}.$$

Induct on the number of boxes. The base case is easy (just consider $n = 0$ or 1 and check that the hook-length formula works in those cases). For the inductive step, assume that the hook length formula holds up to $(n - 1)$ boxes. Then we wish to show that

$$\frac{n!}{H(\lambda)} = \sum_{\nu \text{ corner of } \lambda} \frac{(n-1)!}{H(\lambda-\nu)},$$

where $H(\lambda)$ denotes the product of the hook lengths in the Young diagram λ . This is the same as wanting to show that (dividing through by the left-hand side)

$$1 = \sum_{\nu \text{ corner}} \frac{1}{n} \frac{H(\lambda)}{H(\lambda-\nu)}.$$

We now have a 1 on the left side, and we have a bunch of nonnegative values on the right side. So we can think of this as a probability distribution, and we wish to construct a random process so that the terms in the sum are disjoint probabilities.

For this, consider the following process: first pick any box of λ uniformly at random; call it u . At each subsequent step, jump from u to any other square in the hook of u with equal probability. Repeat this process repeatedly, and stop once we reach a corner ν . Define $p(\nu)$ to be the probability that a hook-walk does end at corner ν .

Claim 45. We have $p(\nu) = \frac{1}{n} \frac{H(\lambda)}{H(\lambda-\nu)}$.

This would give us the desired result, because the random walk must stop at exactly one of the corner vertices. To show this claim, let $P(u, \nu)$ be the probability that a hook walk (u, u', u'', \dots, ν) starting at box u ends at corner ν . This is just summing over all hook walks:

$$P(u, \nu) = \sum_{\substack{\text{hook walk} \\ u \rightarrow u' \rightarrow \dots \rightarrow \nu}} P(\text{hook walk}) = \sum_{u \rightarrow u' \rightarrow \dots \rightarrow \nu} \left(\frac{1}{h(u)-1} \cdot \frac{1}{h(u')-1} \cdots \right).$$

Here's the key observation: when we fix ν , the whole hook-walk stays within the **rectangle** between the top-left corner u and the bottom-right corner ν . Whenever we have a rectangle with corners a, b, c, d (a in the top left and d a

corner), $h(a) + h(d) = h(b) + h(c)$, and similarly

$$(h(a) - 1) + (h(d) - 1) = (h(b) - 1) + (h(c) - 1).$$

The idea is that each hook walk comes with a product of weights $\frac{1}{h(u)-1}$ on vertices, and we have a nice additive identity with inverse weights, so there is lots of potential for simplification. In particular, if d is a corner, $h(d) - 1 = 0$.

So now (restricting to the rectangle of interest) consider a rectangular grid of length $k + 1$ by $\ell + 1$. Let's say the weights of the last row are $\frac{1}{x_1}, \frac{1}{x_2}, \dots, \frac{1}{x_k}$, and B , and the weights in the last column are $\frac{1}{y_1}, \frac{1}{y_2}, \dots, \frac{1}{y_\ell}, B$. (These denominators are hook lengths minus one, and the $\frac{1}{0}$ weight doesn't matter because the bottom-right corner is a vertex and that weight will never be counted in a hook-walk.) Specifying this last row and last column is enough to fill in all of the weights in the rectangle using the relation $(h(a) - 1) + (h(d) - 1) = (h(b) - 1) + (h(c) - 1)$ from above, and call the vertex in the top left corner A :

$\frac{1}{x_1+y_1}$	$\frac{1}{x_2+y_1}$	$\frac{1}{x_3+y_1}$	$\frac{1}{x_4+y_1}$	$\frac{1}{y_1}$	
$\frac{1}{x_1+y_2}$	$\frac{1}{x_2+y_2}$	$\frac{1}{x_3+y_2}$	$\frac{1}{x_4+y_2}$	$\frac{1}{y_2}$	
$\frac{1}{x_1}$	$\frac{1}{x_2}$	$\frac{1}{x_3}$	$\frac{1}{x_4}$	B	

We claim that if we sum the weights of **lattice paths** from A to B (so that we must move by exactly one step each time), the sum of weights is $\frac{1}{x_1 x_2 \dots x_k y_1 \dots y_\ell}$. For example, for a 2 by 2 rectangle, the weights in the boxes will be $\frac{1}{x_1+y_1}, \frac{1}{y_1}, \frac{1}{x_1}$, and 1, so the total sum of hook-walk weights is

$$\frac{1}{x_1 + y_1} \cdot \frac{1}{y_1} + \frac{1}{x_1 + y_1} \cdot \frac{1}{x_1} = \frac{1}{x_1 y_1}.$$

We'll prove this and show the final steps of the main proof next time!

□

6 February 19, 2019

Last time, we were proving the hook length formula – recall that we had variables $x_1, \dots, x_k, \dots, y_1, y_\ell$ that gave us weights on various boxes, and we constructed a $k + 1$ by $\ell + 1$ grid such that every vertex has a weight $w(v)$ with the following rules:

- The bottom row has weights $\frac{1}{x_1}, \frac{1}{x_2}, \dots, \frac{1}{x_k}, ?$.
- The right column has weights $\frac{1}{y_1}, \frac{1}{y_2}, \dots, \frac{1}{y_\ell}, ?$.
- Any other grid square in the i th column and j th row has weight $\frac{1}{x_i+y_j}$, where $i \leq k$ and $j \leq \ell$.

We wanted to consider all lattice paths from A (top left corner) to B (bottom right corner), where each path P has a weight

$$w(P) = \prod_{v \in P} w(v).$$

The following lemma is what we stated last lecture and what we'll need for the proof:

Lemma 46

The sum of all **lattice paths** with the setup above is

$$\sum_{P:A \rightarrow B} w(P) = \frac{1}{x_1 x_2 \cdots x_k y_1 y_2 \cdots y_\ell}.$$

Proof. We induct on $k + \ell$. Base case is $k + \ell = 2$: clearly the vertices have weights $\frac{1}{x_1 + y_1}, \frac{1}{y_1}, \frac{1}{x_1}, 1$, and

$$\frac{1}{x_1 + y_1} \cdot \frac{1}{x_1} + \frac{1}{x_1 + y_1} \cdot \frac{1}{y_1} = \frac{1}{x_1 y_1}.$$

Now any path either goes down from A (to A') by one step, or right from A (to A'') by one step. Thus the sum over all paths of $w(P)$ is

$$\frac{1}{x_1 + y_1} \left(\sum_{P':A' \rightarrow B} w(P') + \sum_{P'':A'' \rightarrow B} w(P'') \right),$$

and by the inductive hypothesis, this is

$$= \frac{1}{x_1 + y_1} \left(\frac{1}{x_1 x_2 \cdots x_k y_2 \cdots y_\ell} + \frac{1}{x_2 \cdots x_k y_1 y_2 \cdots y_\ell} \right),$$

which simplifies, as desired, to

$$\frac{1}{x_2 \cdots x_k y_2 \cdots y_\ell} \cdot \frac{1}{x_1 + y_1} \left(\frac{1}{x_1} + \frac{1}{y_1} \right) = \frac{1}{x_1 \cdots x_k y_1 \cdots y_\ell}.$$

□

Remember that in our hook-walks, we're allowed to skip over rows and columns (as long as we still jump to a spot in our hook), so this is not exactly what we need. So we must modify our proof slightly:

Lemma 47

When we have hook-walks P , the sum of all hook-walks with the setup above (starting at any vertex in the rectangle) is

$$\sum_{P \text{ hook-walk}} w(P) = \left(1 + \frac{1}{x_1}\right) \left(1 + \frac{1}{x_2}\right) \cdots \left(1 + \frac{1}{x_k}\right) \left(1 + \frac{1}{y_1}\right) \cdots \left(1 + \frac{1}{y_\ell}\right).$$

Proof. Imagine expanding out the product fully. Each term corresponds to a subgrid: we skip certain rows and columns if we use a 1 instead of the corresponding $\frac{1}{x_i}$ or $\frac{1}{y_j}$. (Recall that we do not need to start at the top left corner, so we do not need to necessarily include $\frac{1}{x_1}$ and $\frac{1}{y_1}$.) Since any hook-walk is a lattice path on a particular subgrid, this shows the result by Lemma 46. In particular, the factor of $1 \cdot 1 \cdots 1$ just starts at B and finishes immediately. □

We're now ready to finish the proof of the hook length formula from last lecture:

Proof of Theorem 32, continued. Recall that we denote $P(u, v)$ to be the probability that a hook-walk ends up at the corner box v of the Young diagram if we start at u . Any path that ends up at v stays in the rectangular grid between the top left corner and v , so we have

$$P(u, v) = \sum_{p:u \rightarrow v} \frac{1}{h(u) - 1} \frac{1}{h(u') - 1} \frac{1}{h(u'') - 1} \cdots$$

Remember that these weights work out the same way as our grid, so we can apply Lemma 47 (but we need to sum over the allowed starting vertices u just as the lemma does). Fix a corner box v , and call the set of boxes above or to the left of v the “co-hook at v .” These boxes correspond to the x_i s and y_j s in our lemma, so we find that

$$\sum_u P(u, v) = \prod_{t \text{ box in the cohook of } v} \left(1 + \frac{1}{h(t) - 1}\right) = \prod_t \frac{h(t)}{h(t) - 1}.$$

But this product can be rewritten in a suggestive way – if we remove v from the Young diagram, the product of hook lengths $H(\lambda)$ is very similar to before, but now all hook lengths in the co-hook decrease by one! We therefore arrive at the relation

$$\sum_u P(u, v) = \frac{H(\lambda)}{H(\lambda - v)}.$$

The hard work has now been done. To finish, notice that for any fixed box u of λ ,

$$\sum_{v \text{ corner}} P(u, v) = 1,$$

since any hook walk ends at some corner. So now adding over all $u \in \lambda$ and v corner boxes,

$$\sum_{u \in \lambda} \sum_{v \text{ corner}} P(u, v) = n,$$

since there are n boxes. But now switch the order of summation:

$$\sum_{v \text{ corner}} \sum_{u \in \lambda} P(u, v) = n,$$

and by the calculation above, we can plug in the value of $\sum_u P(u, v)$ to find that

$$\sum_{v \text{ corner}} \frac{H(\lambda)}{H(\lambda - v)} = n.$$

Finally, multiplying both sides by $(n - 1)!$ and rearranging gives us

$$\frac{n!}{H(\lambda)} = \sum_{v \text{ corner}} \frac{(n - 1)!}{H(\lambda - v)},$$

which is exactly the recurrence relation we were trying to prove, and we’re done. □

Apparently we can think of everything we’re doing as a “linear extension of a poset,” so we’ll change topics and up the abstraction a bit.

Definition 48

A **poset** or **partially ordered set** \mathcal{P} is a set of objects with a **binary operation** \leq , satisfying the following axioms:

- $a \leq a$ for all $a \in \mathcal{P}$.
- If $a \leq b$ and $b \leq a$, then $a = b$.
- If $a \leq b$ and $b \leq c$, then $a \leq c$.

Also, say that $a < b$ if $a \leq b$ and $a \neq b$.

Note that it is not necessary for either of $a \leq b$ or $b \leq a$ to be true (in other words, unlike something like the real numbers, we don’t need to be able to compare any two elements).

Definition 49

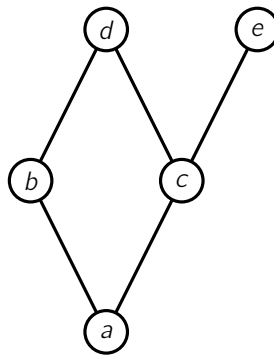
For a, b, c in a poset \mathcal{P} , a is **covered** by b (denoted $a \lessdot b$) if $a < b$ and there is no c such that $a < c < b$.

If we're talking about finite posets, this relation is all we need to know to specify our poset (which elements cover other ones). But infinite posets don't quite work in the same way: there are no covering relations for the real numbers because there is always a real number between any two distinct reals.

Definition 50

The **Hasse diagram** is the graph formed by covering relations, where larger vertices are on the top and smaller vertices are on the bottom.

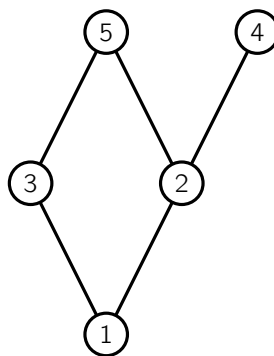
For example, in the diagram here, we have $a \lessdot b, a \lessdot c, b \lessdot d, c \lessdot d, c \lessdot e$, and two elements x, y satisfy $x \leq y$ if there is a path from x to y consisting of upward edges.



Definition 51

A **linear extension** of a finite poset \mathcal{P} with n elements is a bijective map $f : \mathcal{P} \rightarrow \{1, 2, \dots, n\}$ such that $a \leq b \implies f(a) \leq f(b)$.

For example, one potential linear extension of the poset above assigns the following values to the corresponding elements:



Definition 52

The number of linear extensions of \mathcal{P} is denoted $\text{ext}(\mathcal{P})$.

It turns out that we can convert Young tableaux into linear extensions by replacing boxes with dots, connecting two dots if they came from adjacent boxes, and rotating the top left corner to the bottom (135° counterclockwise). Thus, SYTs are special cases of linear extensions. While we do have an expression for the number of SYTs, there isn't a similarly nice one for posets or linear extensions. However, here is an example of a hook-length-type formula that does exist for certain posets:

Theorem 53 (“Baby” hook length formula for rooted trees)

Given a rooted tree T with n nodes, we can correspond it to a poset \mathcal{P}_T . Then

$$\text{ext}(\mathcal{P}_T) = \frac{n!}{\prod_{a \in T} h(a)},$$

where $h(a)$ is the number of nodes including and above a vertex a .

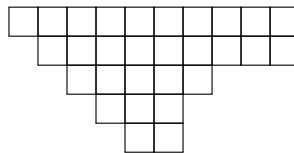
This is an exercise left for us, but it's easier to prove than the other hook length formula.

7 February 20, 2019

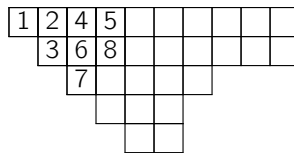
Last time, we introduced Hasse diagrams of posets, and we found that the number of such labeled diagrams (corresponding to linear extensions) for rooted trees is

$$\text{ext}(T) = \frac{n!}{\prod_{a \in T} h(a)},$$

where $h(a)$ is the number of nodes in or downstream of a . We'll start today by introducing one more “hook-length-type formula,” corresponding to **shifted Young diagrams**, which are like Young diagrams but look like



instead of being left-justified. Counting the number of boxes, this corresponds to a partition $\lambda = (10, 9, 5, 3, 2)$, and it now gives the number of ways to partition a number n into **distinct** parts! To get a **shifted Young tableau**, we fill in the shifted Young diagram in the same way, with numbers increasing by row and by column.



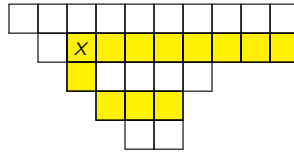
Theorem 54 (Thrall, 1952)

The number of shifted Young tableaux with shifted shape λ with n boxes is similarly

$$\frac{n!}{\prod_{a \in \lambda} h(a)}$$

where $h(a)$, the hook length, now includes a “broken leg” as shown below.

For example, here's a hook with a broken leg for the square x :



Basically, if the hook reaches the left staircase (not the bottom or the right part), it bends over and continues. A proof of this, given by Sagan in 1980, also uses a similar idea of a “hook-walk,” and we can search that up as well. But the point is that there is a nice number for the number of linear extensions in **some** posets, and the proofs are nice as well.

It's time to move on to q -**analog**s, our next topic. The q here serves as a variable but also stands for “quantum;” the idea is that we can have classical and quantum versions of a mathematical object, but as we take $q = 1$ in the quantum version, we get the classical limit. For example, statistical mechanics sometimes has expressions like $q = e^{\hbar\omega/(k_B T)}$, and as we take $\hbar \rightarrow 0$ or $T \rightarrow \infty$, we basically get $q = 1$.

Here are some more generic examples of q -analog

s:

Classical	Quantum
n	$[n]_q = [n] = 1 + q + q^2 + \dots + q^{n-1} = \frac{1-q^n}{1-q}$
$n! = 1 \cdot 2 \cdot \dots \cdot n$	$[n]_q! = [n]! = [1] \cdot [2] \cdot \dots \cdot [n]$

This q -factorial gives us a corresponding q -binomial coefficient as well:

Classical	Quantum
$\binom{n}{k}$	$\begin{bmatrix} n \\ k \end{bmatrix}_q \equiv \begin{bmatrix} n \\ k \end{bmatrix} = \frac{[n]_q!}{[k]_q! [n-k]_q!}$

Example 55

If we use ordinary numbers, we have that $\binom{4}{2} = 6$. But if we use the q -factorial instead, we find that

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix}_q = \frac{[4] \cdot [3]}{[1] \cdot [2]} = \frac{(1+q+q^2+q^3)(1+q+q^2)}{(1+q)} = (1+q^2)(1+q+q^2) = 1+q+2q^2+q^3+q^4.$$

We can make a few observations about these q -binomial coefficients, which we'll list below

- $\begin{bmatrix} n \\ k \end{bmatrix}_q$ is a polynomial in q (of the form $a_0 + a_1q + \dots + a_mq^m$).
- The coefficients a_i are all positive integers, called the **Gaussian coefficients**.
- The coefficients are symmetric or **palindromic**: writing them out backwards gives back the same sequence.
- The coefficients first increase and then decrease: $a_0 \leq a_1 \leq \dots \leq a_{\lfloor m/2 \rfloor} \geq \dots \geq a_m$.

Since ordinary binomial coefficients form a Pascal's triangle, it's natural to ask whether something similar occurs here. The first few q -binomial coefficients are listed below:

$$\begin{array}{cccccccc}
n = 0: & & & & & & & 1 \\
n = 1: & & & & 1 & & & 1 \\
n = 2: & & & 1 & & 1 + q & & 1 \\
n = 3: & 1 & & 1 + q + q^2 & & 1 + q + q^2 & & 1 \\
n = 4: & 1 & 1 + q + q^2 + q^3 & & 1 + q + 2q^2 + q^3 + q^4 & & 1 + q + q^2 + q^3 & 1
\end{array}$$

In the ordinary Pascal's triangle, we have the recurrence relation

$$\binom{n}{r} + \binom{n}{r+1} = \binom{n+1}{r+1}.$$

A similar relation can be observed in the q -analog (which reduces to the ordinary statement if we take $q \rightarrow 1$):

Proposition 56 (q -Pascal's recurrence relation)

For any n, k ,

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \begin{bmatrix} n-1 \\ k \end{bmatrix}_q + q^{n-k} \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_q.$$

Proof. Write out all expressions as factorials:

$$\begin{bmatrix} n-1 \\ k \end{bmatrix}_q + q^{n-k} \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_q = \frac{[n-1]!}{[k]![n-k-1]!} + q^{n-k} \frac{[n-1]!}{[k-1]![n-k]!}.$$

Combining common terms in the numerator and denominator yield

$$= \frac{[n-1]!}{[k]![n-k]!} ([n-k] + q^{n-k}[k]),$$

and now note that $[n-k] = 1 + q + q^2 + \dots + q^{n-k-1}$, and $q^{n-k}[k] = q^{n-k}(1 + \dots + q^{k-1}) = q^{n-k} + \dots + q^{n-1}$. So summing these up together yields $1 + q + \dots + q^{n-1} = [n]$, and thus we simplify to

$$= \frac{[n-1]!}{[k]![n-k]!} [n] = \begin{bmatrix} n \\ k \end{bmatrix}_q,$$

as desired. □

While our calculations so far have been algebraic, it's now time to formulate their combinatorial interpretation. Let λ be a Young diagram, and let the notation $\lambda \subseteq k \times (n-k)$ indicate that the standard (straight) Young diagram fits inside a k by $n-k$ rectangle. In other words, there are at most k nonzero parts in the partition λ , and each part is at most $n-k$:

$$n-k \geq \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k \geq 0.$$

Note that the number of standard Young diagrams that fit within this rectangle is $\binom{n}{k}$, because the interface between boxes and empty spaces is a lattice walk from the bottom left to top right corner. It turns out there is a q -analog of this counting statement:

Theorem 57

For any $0 \leq k \leq n$, we have the recurrence relation

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \sum_{\lambda \subseteq k \times (n-k)} q^{|\lambda|}.$$

For example, for $n = 4, k = 2$, there are 6 possible Young diagrams that fit within a 2×2 box: 1 of them has 0 squares, 1 of them has 1 square, 2 of them have 2 squares, 1 of them has 3 squares, and 1 of them has 4 squares.

And indeed, $\begin{bmatrix} 4 \\ 2 \end{bmatrix}_q = 1 + q + 2q^2 + q^3 + q^4$.

Note that this immediately implies the first three observations about q -binomial coefficients that we had above! In particular, the degree of $\begin{bmatrix} n \\ k \end{bmatrix}_q$ is $k(n - k)$, and palindromicity of the coefficients comes by taking the complement of the shape formed by a Young diagram of x boxes to get a Young diagram of $k(n - k) - x$ boxes.

There's an interesting way to prove the theorem, but we'll instead present the brute-force method:

Proof. Induct on the value of n ; the base case is not hard to prove. We have already established the q -Pascal relation in Proposition 56, so it suffices to show that the same recurrence relation holds for $\sum q^{|\lambda|}$, the right-hand side of this theorem. To do so, consider the first row of the Young diagram – we know $\lambda_1 \leq n - k$ by our constraints.

- If $\lambda_1 < n - k$, meaning $\lambda_1 \leq n - k - 1$, then λ fits inside a k by $n - k - 1$ rectangle (because all future rows must also have at most $n - k - 1$ elements). This case gives the $\begin{bmatrix} n - 1 \\ k \end{bmatrix}_q$ term.
- Otherwise, $\lambda_1 = n - k$, which means the first row is completely filled. Deleting this first row, we get a Young diagram λ contained inside a $k - 1$ by $n - k$ rectangle, and each of these contributes to the right-hand sum like an original Young diagram, but with an additional $n - k$ squares. That's why those terms gain an additional factor of $q^{n-k} \binom{n-1}{k-1}$.

Adding these together shows the q -Pascal relation and thus the stated recurrence relation, as desired. \square

While this proof is short, it does not really explain where the Young diagrams come up or how the algebraic structure arises. So we'll discuss that more next time!

8 February 22, 2019

Our first problem set is posted, and it is due on March 4th. There are 15 problems, but we only have to do about 6 of them to complete the assignment. Later on in the class, we will have one lecture to discuss the problem set solutions.

Today, let's talk more about q -binomial coefficients. Recall that we have the ordinary binomial theorem

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}.$$

Unfortunately, q -binomial coefficients are less nice, especially since in a "quantum world," x and y may not always commute. Treating q as a constant that commutes with both x and y , but setting yx and xy to be different by a factor of q , yields the following q -binomial theorem:

Proposition 58

Suppose that $yx = qxy$, $qx = xq$, and $qy = yq$. Then for any $n \geq 0$, we have

$$(x + y)^n = \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n-k}.$$

For example, taking $n = 2$,

$$(x + y)^2 = (x + y)(x + y) = x^2 + yx + xy + y^2 = x^2 + (1 + q)xy + y^2,$$

and $1, 1 + q, 1$ are indeed the q -binomial coefficients for $n = 2$. This is not too difficult to prove with the q -Pascal relation and induction directly.

One less-clearly-seen relation is Theorem 57 from last lecture, for which we'll try to find a more conceptual proof today. Since the class is called *algebraic combinatorics*, we'll bring in some linear algebra here:

Definition 59

The **Grassmannian** Gr_{kn} is the space of k -dimensional linear subspaces of an n -dimensional space.

More concretely, elements of the Grassmannian can be defined by matrices in the following way. First, pick some reference basis of our ambient n -dimensional space, and pick k linearly independent basis elements for our subspace. Those k basis elements can be written down as n -component vectors, and we can encode those k vectors in a $k \times n$ matrix. However, performing row operations on this matrix performs a change of basis for our vectors without changing the subspace, so that yields the following result:

Proposition 60

The Grassmannian Gr_{kn} is the set of $k \times n$ matrices of rank k , modulo row operations.

It makes sense to ask about the size of this Grassmannian, which is not a well-formed question yet because the entries of the matrix can be real, complex, or values of any finite field. We'll think about the last of these three cases for our purpose:

Fact 61

For any prime p and any natural number r , there is a unique finite field (up to isomorphism) of order $q = p^r$, which we denote \mathbb{F}_q .

We can now restate our problem more concretely:

Problem 62

What is the number of entries in $Gr_{kn}(\mathbb{F}_q)$?

We'll count this in two different ways, and in both cases we'll see the q -binomial coefficient pop up:

Solution 1. Note that row operations correspond to multiplication by $k \times k$ invertible matrices, so we can calculate the number of $k \times n$ matrices of rank k , divided by the number of $k \times k$ invertible matrices, to get our answer.

Any such matrix $A \in \text{Gr}_{kn}(\mathbb{F}_q)$ has rows v_1, v_2, \dots, v_k , where v_i are all vectors in \mathbb{F}_q^n . To make sure A is invertible, all v_k must be independent, and we can guarantee this by picking the vectors successively. First, the number of ways to pick v_1 is $q^n - 1$ (it can be anything except the zero vector). Next, v_2 can be anything except multiples of v_1 , so there are $q^n - q$ ways to pick v_2 . Similarly, there are $q^n - q^2$ ways to pick v_3 , and so on. Overall, the total number of $k \times n$ matrices of rank k is

$$(q^n - 1)(q^n - q) \cdots (q^n - q^{k-1}).$$

But we need to divide by the number of $k \times k$ invertible matrices (for which we can do the same thing but replace n with k), so our total expression is just

$$\frac{(q^n - 1)(q^n - q) \cdots (q^n - q^{k-1})}{(q^k - 1)(q^k - q) \cdots (q^k - q^{k-1})}.$$

Performing a tiny bit of simplification, this is exactly the definition of the q -binomial coefficient $\begin{bmatrix} n \\ k \end{bmatrix}_q$! While this argument only works when q is a prime power, we'll see later that this is enough for our purposes. \square

For our second proof, we'll use another idea from linear algebra:

Solution 2. We transform A into **reduced row-echelon form** with Gaussian elimination.

Remark 63. Row reduction was developed by Newton in 1670, then also Gauss in 1810, and also Jordan in 1888, but also it was known to Chinese mathematicians in 179 AD or even potentially 150 BC. So this is not a new technique in mathematics – it's been around for a long time.

As a review, the general idea of row reduction (when a matrix has at least as many columns as rows) is as follows: find the first nonzero column of the matrix, and use row operations until this column has a 1 in the first row and 0s everywhere else. (We can do this by swapping the rows until the top one has a nonzero entry c , then multiplying that first row by the inverse c^{-1} , and then subtracting off multiples of the top row from the other rows.) Once this is done, we move on to the next column, either clearing all entries in the lower rows or starting the simplification process from the next row. For example, for $k = 5, n = 10$, an example of a reduced form matrix is shown below:

$$\begin{bmatrix} 1 & * & 0 & * & * & 0 & 0 & * & 0 & * \\ 0 & 0 & 1 & * & * & 0 & 0 & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \end{bmatrix}$$

In row reduction language, the 1s are called **pivots**. Those pivot columns are always look the same, and their locations dictate the number of *s that are allowed in subsequent columns. So we can just remove them from consideration without losing any information, and that yields a matrix of the following form:

$$\begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}$$

One student once said that this looks like an American flag, but it is really meant to look like a Young diagram instead. The point is that summing over all pivots gives us a sum over all Young diagram shapes, which is a sum over tableaux

(not Young tableaux) that fit in an $k \times (n - k)$ rectangle! Since each of the *s can be any of the q values in our matrices, we have the following result:

Theorem 64

The number of elements in the Grassmannian $\text{Gr}_{kn}(\mathbb{F}_q)$ is

$$\sum_{\lambda \subseteq k \times (n-k)} q^{|\lambda|}.$$

□

Combining the answers from our two solutions, we've showed the identity Theorem 57 that we want for any power of a prime q . To finish, we go back to Euclid:

Fact 65 (300 BC, Euclid)

There are infinitely many prime numbers.

(Proof: assume finitely many, multiply all together and add 1. This new number is not divisible by any primes, contradiction.) We also use a fact from algebra:

Fact 66

If two rational expressions with integer coefficients $f(q)$ and $g(q)$ are identical on infinitely many values, they are the same on all values of q .

Together, these facts show that Theorem 57 can really be viewed as a statement about elements of the Grassmannian over a finite field, and we now have a more intuitive way to explain the origin of the Gaussian coefficients. The only observation from last lecture left for us to prove is unimodality, but we'll do that later.

For now, we'll continue to explore q -analogs and identities we can write for them, moving towards statistics of permutations. The following fun fact shows a potentially unexpected connection:

Definition 67

For a permutation $w = (w_1, \dots, w_n) \in S_n$, (i, j) is an **inversion** of w if $i < j$ and $w_i > w_j$. Let $\text{inv}(w)$ be the number of inversions of w .

Example 68

$\text{inv}(312675) = 4$, because we have the inversions 31, 32, 65, 75.

Theorem 69

For any n , we have

$$[n]_q! = \sum_{w \in S_n} q^{\text{inv}(w)}.$$

Comparing this to Theorem 57, we see that the size of the Young diagram $|\lambda|$ plays the same role as the number of inversions in w ! But we'll see that there's something more general going on in the future.

Proof. This can be quickly shown by induction on n . The base case is routine as usual, and for the inductive step, we consider how to grow a permutation from a smaller one. Start with any permutation $u \in S_{n-1}$. To get to a permutation in S_n , we can add n in n different places. If it goes at the end, it adds no inversions; if we add it second-to-last, we get 1 inversion; and so on. So we gain a factor of $(1 + q + \dots + q^{n-1}) = [n]$ for each element of S_{n-1} , so the total sum for n becomes $[n] \cdot [n-1]_q! = [n]_q!$, as desired. \square

9 February 25, 2019

Remark 70. *If we have questions about the problem set, we can ask Professor Postnikov! The official office hours are right after lectures on Mondays, but we can also schedule other times if necessary. Also, a few bonus problems will also be added to make the problem set more interesting.*

Last week, we talked about q -binomial coefficients and q -factorials, which are special cases of a more general object:

Definition 71

For any $n = n_1 + \dots + n_r$ with $n_i \geq 0$, the **q -multinomial coefficients** are

$$\left[\begin{matrix} n \\ n_1, n_2, \dots, n_r \end{matrix} \right]_q = \frac{[n]_q!}{[n_1]_q! [n_2]_q! \dots [n_r]_q!}.$$

Notice that (just like for ordinary multinomial coefficients) $\left[\begin{matrix} n \\ 1, 1, \dots, 1 \end{matrix} \right]_q$ is just $[n]_q!$, and $\left[\begin{matrix} n \\ r, n-r \end{matrix} \right]_q$ is just $\left[\begin{matrix} n \\ r \end{matrix} \right]_q$.

Definition 72

A **multiset** is like a regular set, but where entries are allowed to appear multiple times. If the numbers 1 through r appear in S with multiplicities n_1 through n_r , respectively, then we write

$$S = \{1^{n_1}, 2^{n_2}, \dots, r^{n_r}\}.$$

Our first result today will generalize Theorem 69 by now considering permutations $w = (w_1, \dots, w_n)$ on the multiset S :

Definition 73

An **inversion** for a permutation on a multiset w is a pair of indices (i, j) , where $1 \leq i < j \leq n$ and $w_i > w_j$. (Like before, let $\text{inv}(w)$ be the number of inversions in w .)

Then the main theorem is very similar:

Theorem 74

For any q -multinomial coefficient, we have

$$\left[\begin{matrix} n \\ n_1, \dots, n_r \end{matrix} \right]_q = \sum_w q^{\text{inv } w},$$

where the sum is taken over all permutations of $\{1^{n_1}, 2^{n_2}, \dots, r^{n_r}\}$.

This proof can be similarly done by induction, and this is one of the problems we may attempt on our problem set. And as a corollary, we know that $\begin{bmatrix} n \\ n_1, \dots, n_r \end{bmatrix}_q$ is always a polynomial in q with positive integer coefficients (because the right-hand side of our theorem is). The degree d of this polynomial is the maximum number of inversions in any permutation of $\{1^{n_1}, 2^{n_2}, \dots, r^{n_r}\}$, which happens when we write the elements of S in weakly decreasing order. Thus,

$$d = \sum_{1 \leq a < b \leq r} n_a n_b.$$

Additionally, we also know that the coefficients of the polynomial are symmetric (meaning $a_i = a_{d-i}$), just like in the binomial coefficient case. This is because

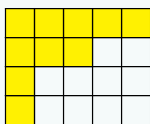
$$\text{inv}(w_1, \dots, w_n) = d - \text{inv}(w_n, w_{n-1}, \dots, w_1),$$

since any pair of **distinct** entries ($w_i \neq w_j$) is an inversion in either (w_1, \dots, w_n) or (w_n, \dots, w_1) , and d is the total number of pairs of **distinct** entries.

Finally, to discuss one more special situation, consider the case where our multiset only contains 1s and 2s, so $S = \{1^k 2^{n-k}\}$. Then there is a correspondence between permutations of S and Young diagrams $\lambda \subseteq k \times (n-k)$, which we can see illustrated by example:

Example 75

Let $w = (2, 1, 1, 2, 2, 1, 2, 2, 1)$, which is a permutation of the set $\{1^4 2^5\}$. We may transform this into a lattice path on a 4×5 rectangle of squares, going up when we see a 1 and right when we see a 2. The path is traced out below, and the yellow region is our Young diagram:



Under this visualization, the **number of squares** $|\lambda|$ corresponds to the **number of inversions**, since each yellow square corresponds to a 2 showing up before a 1! (And if we try to tell the same story for $r = 3$, we can think of a permutation as a lattice path in a 3-dimensional box, where we go up, right, or into the page each time we see a 1, 2, 3 respectively. But things don't work out quite as nicely.)

That's all we'll say about q -analogs for now, so we'll move on to our next topic. Let $[n]$ denote the set $\{1, 2, \dots, n\}$, and let w be a permutation of that set (which we can more abstractly think of as a **bijective map** $w : [n] \rightarrow [n]$). We can compose two such bijective maps, which is the way we'll define **multiplication** of permutations. Under this multiplication, these permutations form a group S_n , called the **symmetric group**. (Note that Stanley's book uses the notation \mathfrak{S}_n instead of S_n .)

There's several different ways we can notate permutations. The first three can be compared as shown below:

Name	Notation	Example
1-line notation	(w_1, \dots, w_n)	$(2, 5, 7, 3, 1, 6, 8, 4)$
2-line notation	$\begin{pmatrix} 1 & 2 & \dots & n \\ w_1 & w_2 & \dots & w_n \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 5 & 7 & 3 & 1 & 6 & 8 & 4 \end{pmatrix}$
cycle notation	$(a_1 a_2 a_3) \dots$	$(125)(3784)(6)$

The first two types of notation here basically notate a bijective map sending i to the i th number that shows up in the permutation. But the third type of notation, cycle notation, is the most important: the idea is that we keep

following the permutation map until we arrive back to a point we've already been at, and that closes off a "cycle" traced out by the map. (Trivial cycles like (6) are sometimes omitted, and they're **fixed points** of the bijective map w .)

There are two other common ways to notate permutations which we'll mention as well. One of them is **graphical notation**, where we draw n nodes containing the numbers $\{1, 2, \dots, n\}$ and draw an edge from i to the i th element of the permutation (so that the cycles in cycle notation form closed polygons). Finally, in the vein of multiplying different bijective maps (or thinking about representation theory), we have **matrix notation**. Matrix notation encodes a permutation in an $n \times n$ matrix $a = (a_{ij})_{i,j=1}^n$, where

$$a_{ij} = \begin{cases} 1, & j = w(i) \\ 0, & \text{otherwise.} \end{cases}$$

For the permutation (2, 5, 7, 3, 1, 6, 8, 4), the matrix is "either this one or its transpose:"

$$w = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

(As an exercise, it's good for us to check whether this one or the other one is correct.) Notice that with matrix notation, permutations correspond to **rook placements** on a chessboard (we place rooks where there are 1s, in a way so that there are no rooks attacking each other). There are interesting problems we can pose about non-attacking rook placements, and we'll talk about them later in this class.

But for now, we're going to discuss the elements of the symmetric group in more detail by analyzing **statistics on permutations**. Basically, all of these statistics can be thought of as some function

$$A : S_n \rightarrow \{0, 1, 2, \dots, \}$$

counting some property of the permutation, and we can then form a generating function for the statistic A :

$$F_A(x) = \sum_{w \in S_n} x^{A(w)}.$$

Definition 76

Two statistics A and B are **equidistributed** if they have the same generating function.

An example of a statistic is the number of inversions $\text{inv}(w)$ of a permutation w that we've already mentioned. We'll spend today's lecture and part of next lecture listing some other ones:

Definition 77

The **length** of w , denoted $\ell(w)$, is the minimum number of **adjacent transpositions** (of the form s_i , switching i and $i + 1$ for any $1 \leq i \leq n - 1$) that we must compose to be equivalent to w .

It's easy to check that we can write any permutation as a composition of adjacent transpositions: induct on n by switching n into the last spot.

Example 78

We can perform the adjacent transpositions $123 \rightarrow 213 \rightarrow 231 \rightarrow 321$, so $\ell(321)$ is at most 3. A bit more brute-force work verifies that the permutation $(3, 2, 1)$ indeed has length 3, and this is also the number of inversions of $(3, 2, 1)$.

It turns out that this is a more general phenomenon:

Theorem 79

For any permutation w , the length of w is also the number of inversions of w .

So those two statistics aren't just equidistributed – they're actually the same statistic. (Indeed, since each adjacent transposition can only add 1 to the number of inversions, and equality is achieved by induction, performing the adjacent transpositions for n last.)

Definition 80

For any permutation w , the number of cycles in w (including fixed points) is denoted $\text{cyc}(w)$.

For example, $w = (2, 5, 7, 3, 1, 6, 8, 4)$ in cycle notation is $(125)(3784)(6)$, so $\text{cyc}(w) = 3$. Note that by “degree arguments,” this can't be equidistributed with the number of inversions (since we can only have at most n cycles but far more inversions). In fact, we can write out the generating function explicitly:

Theorem 81

For any n ,

$$\sum_{w \in S_n} x^{\text{cyc}(w)} = x(1+x)(2+x) \cdots (n-1+x).$$

Proof. We induct on n (the base case $n = 1$ is clear). Write the permutations of S_{n-1} in cycle notation, and let's say we insert n into any of those permutations. Then n can either form its own cycle or join an existing one.

Since there are k places to insert n into a cycle of length k (each leading to a different final cycle), there are $n - 1$ ways to insert into an existing cycle and keep the number of cycles the same, and 1 way to add a new cycle. That gives us the additional $((n - 1) + x)$ factor to multiply into our generating function, as desired. \square

10 February 27, 2019

Remark 82. As a reminder, our first problem set is due on Monday, so we should start it soon if we haven't already done so. A few bonus problems were also added which are a bit more challenging than the original ones.

Next Wednesday, we will discuss the problem set in class. Professor Postnikov is pretty lenient with late problem sets, but we (obviously) can't turn them in after we discuss the solutions.

Last time, we started discussing statistics on permutations. Specifically, we defined $\text{inv}(w)$ to be the number of inversions and $\text{cyc}(w)$ to be the number of cycles, and we found the generating functions

$$\sum_{w \in S_n} q^{\text{inv}(w)} = [n]_q!, \quad \sum_{w \in S_n} x^{\text{cyc}(w)} = x(x+1) \cdots (x+n-1).$$

Definition 83

A **descent** of a permutation $w = (w_1, \dots, w_n)$ is an index $1 \leq i \leq n - 1$ such that $w_i > w_{i+1}$. We denote the number of descents by $\text{des}(w)$.

For example, $\text{des}(2, 5, 7, 3, 1, 6, 8, 4) = 3$ (corresponding to indices 3, 4, 7). It turns out that the generating function for the number of descents is a bit less clean than the one for inversions or cycles but is still interesting

Definition 84

The generating function $\sum_{w \in S_n} x^{\text{des}(w)}$ is the **Eulerian polynomial**.

A meta-mathematical claim is that interesting permutation statistics are likely equidistributed with one of these classes (inversions, cycles, or descents), meaning that their generating function is one of the forms that we've already seen. (Statistics related to inversions are called "Mahonian statistics;" cycles are related to Stirling numbers, but there's no common name for the associated statistics; and descent-related statistics are called "Eulerian statistics.")

Definition 85

The **major index** of a permutation $w \in S_n$ is

$$\text{maj}(w) = \sum_{\substack{i \text{ descent} \\ \text{of } w}} i.$$

For example, the permutation $w = (2, 5, 7, 3, 1, 6, 8, 4)$ with descents in positions 3, 4, and 7 has $\text{maj}(w) = 3 + 4 + 7 = 14$.

Theorem 86

The statistics $\text{inv}(w)$ and $\text{maj}(w)$ are equidistributed.

Proving this is one of the bonus problems on our first problem set!

Remark 87. Both the major index and the "Mahonian statistics" for inversions are named after Major Percy MacMahon, who wrote a famous book on combinatorics (so in other words, the "major" in "major index" refers to the military rank).

Definition 88

A **record** of a permutation w is an entry greater than all entries to the left of it (in one-line notation). We denote the number of records of w by $\text{rec}(w)$.

For example, $w = (2, 5, 7, 3, 1, 6, 8, 4)$ has the records 2, 5, 7, 8, so $\text{rec}(w) = 4$.

Theorem 89

The statistics $\text{rec}(w)$ and $\text{cyc}(w)$ are equidistributed.

This can be proved by induction, for example by showing that the generating functions satisfy the same recurrence relation. But we prefer combinatorial proofs, so we'll show a bijective argument here:

Proof. We construct a bijection $f : S_n \rightarrow S_n$ sending $w \rightarrow \tilde{w}$, such that $\text{cyc}(w) = \text{rec}(\tilde{w})$. To do this, write a permutation w in cycle notation as

$$w = (a_1 \cdots)(a_2 \cdots)(a_3 \cdots) \cdots,$$

where we fix the convention that each a_i is the **maximal element** in its cycle, and sort them such that $a_1 < a_2 < \cdots$. Then the corresponding \tilde{w} is w but viewed in one-line notation without the parentheses. For example,

$$w = (125)(3784)(6) \implies w = (512)(6)(8437) \implies \tilde{w} = (5, 1, 2, 6, 8, 4, 3, 7).$$

Notice that the first elements a_1, a_2, \cdots in the cycles of w will be the records of \tilde{w} , so the number of cycles of w is the number of records of \tilde{w} , as desired. And this map is indeed bijective (because we can find the records in \tilde{w} , mark them as starting points of cycles, and put the parentheses back). Thus, $\text{rec}(w)$ and $\text{cyc}(w)$ are indeed equidistributed. \square

Definition 90

An **exceedance** of a permutation w is an index $1 \leq i \leq n$ such that $w_i > i$. Denote the number of exceedances of w by $\text{exc}(w)$.

For example, in the permutation $(2, 5, 7, 3, 1, 6, 8, 4)$, we have 4 exceedances (not counting 6, which is a “weak exceedance”), so $\text{exc}(w) = 4$.

Theorem 91

The statistics $\text{exc}(w)$ and $\text{des}(w)$ are equidistributed (another way to say this is that “the number of exceedances is an Eulerian statistic”).

Proof. Define an **anti-exceedance** to be an index i such that $w_i < i$. This is equidistributed with the number of exceedances, because we can pair up w and w^{-1} – if $w(i) > i$, then $w^{-1}(w(i)) < w(i)$. Thus, if i is an exceedance in w , then $w(i)$ is an anti-exceedance in w^{-1} , and thus the number of exceedances in w is the number of anti-exceedances in w^{-1} . Showing equidistribution of $\text{exc}(w)$ and $\text{des}(w)$ follows from the following fact:

Claim 92. *If $w \rightarrow \tilde{w}$ is a map that converts cycle notation to one-line notation, then the number of anti-exceedances in w is the number of descents in \tilde{w} .*

To verify this, note that if i is a descent (index) in \tilde{w} , then the $(i+1)$ st entry in \tilde{w} (in one-line notation) is not larger than \tilde{w} , meaning that the i th and $(i+1)$ th indices are in the same cycle of w (because we fixed the notation so that each new cycle starts with a record.) The index i being a descent therefore means that (reading the cycle notation) the i th entry of w is mapped to the $(i+1)$ th entry of w , a smaller element, which makes it an anti-exceedance. This finishes the proof. \square

This is all we’ll say about permutation statistics for now, and we’ll turn our attention to another set of special numbers:

Definition 93

For $0 \leq k \leq n$, define the **Stirling numbers of the first kind**

$$s(n, k) = (-1)^{n-k} c(n, k),$$

where $c(n, k)$ are the **signless Stirling numbers of the first kind**

$c(n, k)$ = the number of permutations in S_n with k cycles.

By convention, let $s(0, 0) = 1$ and $s(n, 0) = 0$ for all $n \geq 1$.

Note that in the above definition, fixed points count as cycles (of length 1).

Theorem 94

The generating functions for the (signless and signed) Stirling numbers are

$$\sum_{k=0}^n c(n, k)x^k = x(x+1)\cdots(x+n-1), \quad \sum_{k=0}^n s(n, k)x^k = x(x-1)\cdots(x-n+1).$$

The former of these two generating functions is called the **raising power of x** , while the latter is called the **falling power of x** , sometimes denoted $(x)_n$.

Definition 95

For $0 \leq k \leq n$, define the **Stirling number of the second kind**

$S(n, k)$ = number of set-partitions of $[n]$ into k non-empty blocks.

We also use the convention $S(0, 0) = 1$ and $S(n, 0) = 0$ for $n \geq 1$.

Unlike the Stirling numbers of the first kind, there are no negative signs ($S(n, k)$ is always positive).

Example 96

An example of a set-partition for $n = 8, k = 3$ is $\pi = (125|3478|6)$; the main difference between this set-partition and cycle notation for permutations is that the order **within** each group doesn't matter.

Theorem 97

We have the relation (similar to a generating function)

$$\sum_{k=0}^n S(n, k)(x)_k = x^n.$$

We may compare this to the generating function for Stirling numbers of the first kind in Theorem 94: there, we input powers of x , and get falling powers of x , but in this one, we input falling powers of x , and get powers of x . This is a kind of duality that we will explore more next time!

11 March 1, 2019

We will continue our discussion of Stirling numbers today. Recall that $s(n, k)$, the Stirling numbers of the first kind, are $(-1)^{n-k}$ times the number of permutations of S_n with exactly k cycles (including fixed points). Meanwhile, $S(n, k)$, the Stirling numbers of the second kind, count the number of set-partitions of $[n]$ into k blocks, where order doesn't matter within each block.

Example 98

Take $n = 4, k = 2$. There are 8 elements of S_n which consist of a 3-cycle and a fixed point, and 3 elements consisting of two transpositions. Thus,

$$s(4, 2) = (-1)^{4-2} \cdot (8 + 3) = 11.$$

The computation for $S(4, 2)$ is similar, but we don't care which orientation the 3-cycle goes in (only which elements it contains). So we have 4 3-cycles and 3 pairs of transpositions, and this means

$$S(4, 2) = 4 + 3 = 7.$$

Last lecture, we stated Theorem 94 and Theorem 97, which encoded the two different types of Stirling numbers in generating-function-like equations. We'll take another perspective on those results today, thinking about the vector space of polynomials $\mathbb{R}[x]$.

Fact 99

The infinite set $\{1, x, x^2, \dots, x^n, \dots\}$ is a basis for $\mathbb{R}[x]$, and so is $\{1, x, x(x-1), \dots, (x)_n, \dots\}$. Because the generating function results write elements of each basis in terms of the others, **Stirling numbers are a "change of basis."**

In particular, we can construct matrices for both changes of bases:

Corollary 100

If we construct change-of-basis (infinite) matrices $(s(n, k))_{n, k \geq 0}$ and $(S(n, k))_{n, k \geq 0}$, then the matrices are inverses of each other.

Example 101

If we restrict the matrices to $0 \leq n, k \leq 3$, then

$$(s(n, k)) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 2 & -3 & 1 \end{bmatrix}, \quad (S(n, k)) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 3 & 1 \end{bmatrix}.$$

Both of these change-of-basis matrices are lower triangular, since $s(n, k) = S(n, k) = 0$ for $k > n$, and we can check that these two 4×4 matrices are indeed inverses.

The generating function results of both Theorem 94 and Theorem 97 can be proven by induction, but we'll now show a combinatorial proof of the latter:

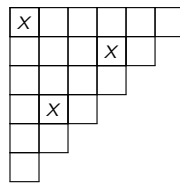
Proof of Theorem 97. It is enough to prove that this identity holds for all positive integers x , since both sides are polynomials. Consider the set of all functions $F = \{f : [n] \rightarrow [x]\}$ – each of $1, 2, \dots, n$ has x options, so there are a total of x^n functions in F .

On the other hand, given a function f , we can construct a set-partition π of $[n]$, putting i and j in the same block if and only if $f(i) = f(j)$. So we group elements of our set based on their output, and we'll count the number of functions that produce a given set-partition by casework. If we fix a set-partition π with k blocks B_1, B_2, \dots, B_k , then the number of functions that produce this specific set-partition is $x(x-1)\cdots(x-(k-1)) = (x)_k$, since B_1 has x options for its corresponding f -value, then B_2 has $(x-1)$ options (since it can't be equal to the value on B_1), and so on. But now we're done: the number of such functions over all set-partitions is

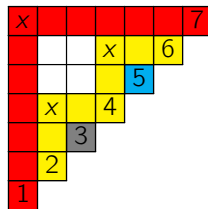
$$|F| = \sum_{k=0}^n S(n, k)(x)_k,$$

since we just pick a set-partition and then assign values to it, and setting this equal to the total size of x^n yields the result. □

We'll now see another combinatorial application of Stirling numbers of the second kind, **rook placements** (in which rooks are placed on a grid or a subset of a grid, with the constraint that they cannot attack each other). Specifically, consider the number of rook placements on a triangle board of Young tableau shape $((n-1), (n-2), \dots, 1)$. An example rook placement for $n = 7$ is shown below:



Now, add boxes in the immediately adjacent squares (below and to the right) with numbers from 1 to n , and for each rook, place a hook that connects two of the bottom-right corners.



We can then correspond a rook placement with a set-partition by having all numbers connected by hooks in a given part of the partition! In the above picture, 2, 4, 6 are connected, 1 and 7 are connected, and 3 and 5 are lonely, so this is the set-partition

$$\pi = (1, 7 \mid 2, 4, 6 \mid 3 \mid 5).$$

Since adding each new rook essentially combines two parts of a set-partition together, we get the following result:

Theorem 102

$S(n, k)$ is the number of non-attacking rook placements on the triangle board above with $n - k$ rooks.

For instance, $S(n, n) = 1$, and this is indeed the number of ways to place $n - n = 0$ rooks down on our triangular board.

Definition 103

The **Bell number** $B_n = \sum_{k=0}^n S(n, k)$ is the total number of set-partitions of n (or equivalently the total number of rook placements on $((n-1), (n-2), \dots, 1)$).

Example 104

For $n = 4$, there is 1 way to place no rooks, 6 ways to place 1 rook, 7 ways to place 2 rooks (do casework on one rook's placement), and 1 way to place 3 rooks. Adding these together yields $B_4 = 15$.

There are other useful ways to categorize set-partitions as well, and one such way comes from drawing **arc diagrams**. To create these, first place the numbers 1 through n on a line segment, and draw an arc between adjacent entries in a partition block (where "adjacent" means that there is no number between them in the partition block). Note that we can connect this back to the rook placement diagram by corresponding an arc from a to b with a rook on the a th column and b th row.

Courtesy of Agustin Garcia, here's the arc diagram for $(1, 4, 6 \mid 2, 3 \mid 5)$:



Definition 105

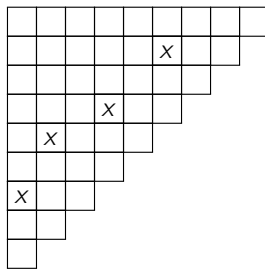
A set-partition is **non-crossing** if no two arcs intersect each other. In other words, if $i < j < k < l$, we can't have an arc between i and k and also between j and l for a non-crossing partition. Similarly, a set-partition is **non-nesting** if there is no arc "inside" another one. In other words, we can't have an arc between i and l and also between j and k for a non-nesting partition.

For instance, the arc diagram above corresponds to a non-crossing but not non-nesting partition. There is an interesting duality between these two definitions:

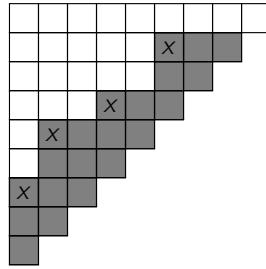
Theorem 106

The number of non-crossing set-partitions of $[n]$ is equal to the number of non-nesting set-partitions of $[n]$, and both of these are equal to the Catalan number C_n .

Proof for non-nesting set-partitions. We'll use the rook placements description of set-partitions. Notice that a set-partition π is non-nesting if and only if there is no rook in the southeast corner of another one. For example, the diagram below shows an example of a valid rook placement:



To obtain a connection to Catalan numbers, we can imagine placing a sun on the top-left corner of our picture, so that every rook casts a shadow below and to its right:



For non-nesting set-partitions, the shadow shape uniquely determines the positions of the rooks (the top-left corner of the shadow). And now following the border of the shadow gives us a Dyck path (with up corresponding to $+1$ and right corresponding to -1), so the number of non-nesting set-partitions is indeed C_n , the number of Dyck paths. \square

Figuring out the analogous argument for non-crossing set-partitions is one of our bonus problems for our problem set. And we'll mention one last fact about these Stirling numbers:

Fact 107

It turns out that there is also a Pascal's triangle-like object for $S(n, k)$. Closely related, there is a Pascal-like recurrence relation for both $S(n, k)$ and $s(n, k)$, and we'll discuss this more next time.

12 March 4, 2019

We'll discuss the Eulerian numbers today:

Definition 108

The **Eulerian numbers** $A_{n,k}$ are the number of permutations $w \in S_n$ with exactly k descents.

(We should also be careful not to confuse these with Euler numbers, another important sequence of integers!) We're going to find a combinatorial connection with these by constructing a nice bijection between **permutations** in S_n and **increasing binary trees** on n nodes. We used Catalan numbers earlier in the class to characterize certain binary trees, but this time we're doing something a bit different: we'll label the vertices by numbers from 1 to n such that they **increase as we go away from the root**.

Example 109

Consider the tree formed by the permutation $w = (4, 2, 8, 5, 1, 3, 9, 10, 6, 7)$.

At the root of this tree, we must use 1, so we'll have the left branch of the tree contain $(4, 2, 8, 5)$ and the right branch contain everything else. Now within each of those groups, the rootmost node will be the smallest element, and we can continue to iterate this process!

So each node has at most 1 left child (the smallest element on its left, if there are elements to its left) and at most 1 right child (the smallest element on its right, again if one exists), which means it is indeed a binary tree. The nodes are labeled 1 through n , and they indeed increase downwards. And this is a bijection, since we can just reverse the process by placing 1 in our permutation, putting the left elements to its left and right elements to its right, and so on.

Theorem 110

$A_{n,k}$ is the number of increasing binary trees on n nodes with k left edges.

Proof. When we construct a tree T_w from a permutation w , any left edge pointing from i to j corresponds to the permutation reading (in one-line notation)

$$w = (\dots, j, \dots, i, \dots),$$

where j is the minimal element within the bracketed part. So the left edge originating from i corresponds to a descent ending at i . Since every descent also corresponds to a left edge (we must split up the two elements in the descent at some point), descents and left edges correspond to each other, as desired. \square

In a sense, Eulerian numbers look a lot like binomial coefficients:

Definition 111

The **Eulerian triangle** is constructed such that the b th entry in the a th row is $A_{a,b-1}$. The first few rows are shown below:

$$\begin{array}{cccccc}
 & & & & & 1 \\
 & & & & 1 & & 1 \\
 & & & 1 & & 4 & & 1 \\
 & & 1 & & 11 & & 11 & & 1 \\
 1 & & 26 & & 66 & & 26 & & 1.
 \end{array}$$

This is a lot like Pascal's triangle, but the weights are different. For example, notice that $26 = 11 \cdot 2 + 1 \cdot 4$ and that can be understood by the fact that 11 is in the second left-slanting diagonal, while 1 is in the fourth right-slanting diagonal. That's what this next result says:

Theorem 112 (Recurrence relation for Eulerian numbers)

For all n, k , we have

$$A_{n+1,k} = (n - k + 1)A_{n,k-1} + (k + 1)A_{n,k}.$$

Proof. We'll prove this result with both perspectives of the Eulerian numbers. First of all, suppose we have a binary tree with n nodes and we want to add an extra node. To do this, we must add an extra leaf with label $n + 1$; it can be checked that if we already have k left edges, then the number of ways to add a left edge is $(n - k + 1)$, and the number of ways to add a right edge is $(k + 1)$ (because each left edge creates a new location to add a right edge, and vice versa).

Alternatively, we can analyze the permutations; suppose we have a permutation $w \in S_n$ and we want to add $n + 1$ to it. Adding it in one of the k descent slots or at the beginning does not add a descent ($k + 1$ ways), while adding it in an ascent slot or at the end adds a descent (the remaining $n - k + 1$ ways, assuming that we started with $(k - 1)$ descents to get to a final of k). \square

Now that we've constructed one Pascal-like triangle, we'll return to the Stirling numbers again and understand their recurrence relations in the same way. First, recall that $c(n, k)$, the signless Stirling numbers of the first kind, are

the number of permutations $w \in S_n$ with k cycles (including fixed points). Below is the beginning of the Pascal-like triangle where the b th entry in the a th row is $c(a, b)$.

			1		
		1		1	
	2		3		1
	6	11		6	1

We no longer have the same symmetry across the vertical axis, but we still have a recurrence relation with weights which looks relatively simple:

Proposition 113 (Recurrence relation for $c(n, k)$)

For all n, k , we have

$$c(n + 1, k) = c(n, k - 1) + nc(n, k).$$

Meanwhile, $S(n, k)$, the Stirling numbers of the second kind, are the number of set-partitions of $[n]$ into k groups. If we construct a similar triangle, it looks like

				1				
			1		1			
		1		3		1		
	1		7		6		1	
	1	15		25		10		1

We again have a recurrence relation, now dependent on k rather than n (such that it is slightly harder to infer directly from the Pascal's triangle):

Proposition 114 (Recurrence relation for $S(n, k)$)

For all n, k , we have

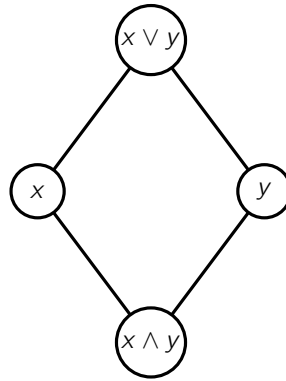
$$S(n + 1, k) = S(n, k - 1) + kS(n, k).$$

Proving these relations are left as exercises for us. There are many more permutation statistics that we could continue to discuss, but we'll move on to the next topic for now: **posets and lattices**.

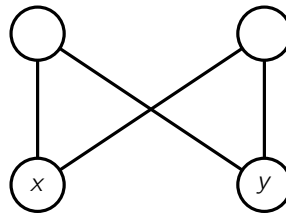
Definition 115 (Poset definition of a lattice)

A **lattice** L is a special kind of poset with two binary operations, **meet** \wedge and **join** \vee . For any $x, y \in L$, $x \wedge y$ is the unique maximal element of L which is less than or equal to both x and y , and $x \vee y$ is the unique minimal element of L such that it is greater than or equal to both x and y .

Below is a simple example of a lattice:



However, here's an example of a poset that is not a lattice, since x and y have no $x \wedge y$ and two different potential $x \vee y$ s (both of which are not allowed):



We would like to encode all of the properties of lattices together in a more formal and abstract way, and that's what we'll do now.

Definition 116 (Axiomatic definition of a lattice)

A **lattice** is a set L with two binary operations \wedge and \vee satisfying the following properties:

- \wedge and \vee are commutative and associative – in other words, for any $x, y, z \in L$, $x \wedge y = y \wedge x$, $x \vee y = y \vee x$, $x \wedge (y \wedge z) = (x \wedge y) \wedge z$, and $x \vee (y \vee z) = (x \vee y) \vee z$,
- for any $x \in L$, $x \wedge x = x \vee x = x$,
- (absorption law) for any $x, y \in L$, $x \wedge (x \vee y) = x = x \vee (x \wedge y)$,
- for any $x, y \in L$, $x \wedge y = x$ if and only if $x \vee y = y$.

Note that addition and multiplication do not follow these laws, so our binary operations are doing something completely different.

Fact 117

Given the axiomatic operations \wedge and \vee on a set L , we can always reconstruct a poset. On the flip side, given any lattice poset, it and its operations \wedge and \vee will always satisfy the axioms of a lattice. In particular, $x \leq y$ if and only if $x \wedge y = x$ (or equivalently $x \vee y = y$).

Our next few lectures will be devoted to student presentations of problems, and then we'll return to discussing special types of lattices.

13 March 6, 2019

We got our problem sets back in class today, and students are presenting selected solutions to those problems.

Problem 118

Prove a permutation is queue-sortable if and only if it is 321-avoiding.

Solution by Fadi Atieh. First, we prove that if a permutation w_1, \dots, w_n has a 321-pattern, then it is not queue-sortable. In this case, there are some indices $i < j < k$ such that $w_i > w_j > w_k$. When we get to index i , we can't put w_i in the list yet (because w_j is smaller than it) so w_i is pushed in the queue. However, w_j also cannot be pushed immediately, so it must go in the queue as well. Thus, w_i will exit the queue before w_j , which is a contradiction.

Now, suppose that w is not queue-sortable. We claim that for any queue-sortable permutation w , there is a **unique** way to queue-sort it. This is our sorting process involves a sorting pointer (which tells us which step we're at), a queue, and a partially filled list. Whenever we get to an element a , if there is something smaller than it that still hasn't been put in the list, we must put a in the queue, and otherwise it goes in the list (or goes in the queue and immediately comes out). Thus, if we get stuck not being able to do either of these things, we must be trying to put some element w_i in the partially-filled list which currently ends in w_k , but $w_i > w_j > w_k$ for some w_j in the queue. This means $w_i > w_j > w_k$, but w_k was in the permutation before w_j , which was before w_i . Thus, we've found a 321-pattern with the indices (i, j, k) as desired. \square

Problem 119

Prove the identity

$$\begin{bmatrix} 2n \\ n \end{bmatrix}_q = \sum_{k=0}^n q^{k^2} \begin{bmatrix} n \\ k \end{bmatrix}_q^2.$$

Solution by Agustin Garcia. The left hand side is the q -generating function for the set of Young diagrams that fit inside $n \times n$ rectangles (recall Example 75). For any such diagram, there is exactly one maximal $k \times k$ box that fits in the upper left hand corner (this is called the **Durfee square** of the Young diagram). This means we can uniquely split our Young diagram into a $k \times k$ square and then two Young diagrams inside $k \times (n - k)$ rectangles, which are each generated by the function $\begin{bmatrix} n \\ k \end{bmatrix}_q$. Tack on a q^{k^2} for the $k \times k$ square, and we have the desired result. \square

Problem 120

Show that the number of set-partitions of $[n]$ such that i and $i + 1$ are not in the same set (for all $1 \leq i \leq n - 1$) is equal to the number of set-partitions of $[n - 1]$.

We'll present two solutions:

Solution by Christina Meng. We'll think about set-partitions as rook placements, as we did in lecture. The left-hand side then tells us that we want to place rooks in a board with rows of $n - 1, n - 2, \dots, 1$ squares, but if we can't have i and $i + 1$ in the same partition, then we can't have any rooks in any of the bottom-right corners (since those would connect two adjacent numbers). Thus, this is equivalent to just having a rook placement on a board with rows $n - 2, n - 3, \dots, 1$, which is what the right-hand side counts. \square

Solution by Sophia Xia. We construct a bijection between the two sets – given a partition π of $[n - 1]$, we want to map this to a partition of $[n]$ with no two consecutive integers in the same block. For this, look at each block in the

partition; for every maximal sequence $i, i + 1, \dots, j$ of consecutive integers in a block of π , remove $j - 1, j - 3, \dots$, until we reach either i or $i + 1$, and place all of them in a block with n . This yields a partition of $[n]$ with no two consecutive integers in the same block, since an element j can only get placed in the partition with n if $j + 1$ is not. To construct the inverse map, place each element k in the partition with n into the partition that contains $(k + 1)$ (which was not moved by the logic above). Thus these two sets have the same number of elements. \square

Problem 121

Show the identity

$$(x + y)^n = \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n-k},$$

where $yx = qxy$, $qx = xq$, $qy = yq$.

Solution by Mathew Ganatra. Let $\begin{bmatrix} 0 \\ 0 \end{bmatrix}_q = 1$, and define $\begin{bmatrix} n \\ k \end{bmatrix}_q = 0$ for $n \in \mathbb{N}$ and $k > n$ or $k < 0$. We will make use of the identity (from class)

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \begin{bmatrix} n-1 \\ k \end{bmatrix}_q + q^{n-k} \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_q.$$

We proceed by induction. The base case $n = 0$ is clear, and for the inductive step, assume the identity holds for all integers $m \leq n$. Then the left-hand side is equal to

$$(x + y)^{n+1} = (x + y)^n(x + y) = \boxed{\sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n-k} (x + y)}.$$

On the other hand, the right-hand side is equal to

$$\sum_{k=0}^{n+1} \begin{bmatrix} n+1 \\ k \end{bmatrix}_q x^k y^{n+1-k} = \sum_{k=0}^{n+1} \left(\begin{bmatrix} n \\ k \end{bmatrix}_q + q^{n+1-k} \begin{bmatrix} n \\ k-1 \end{bmatrix}_q \right) x^k y^{n+1-k}$$

by our identity, and expanding (and removing the $k = 0$ term from the second sum) yields

$$\sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n+1-k} + \sum_{k=1}^{n+1} q^{n+1-k} \begin{bmatrix} n \\ k-1 \end{bmatrix}_q x^k y^{n+1-k}.$$

Now shifting the index of summation, this is

$$\sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n+1-k} + \sum_{k=0}^n q^{n-k} \begin{bmatrix} n \\ k \end{bmatrix}_q x^{k+1} y^{n-k} = \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q (x^k y^{n+1-k} + q^{n-k} x^{k+1} y^{n-k}).$$

Finally, because $q^{n-k} x y^{n-k} = y^{n-k} x$ (by moving the x past the y s one at a time), we can move a y and an x to the end of the two terms, respectively:

$$= \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q (x^k y^{n-k} y + x^k y^{n-k} x) = \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n-k} (x + y),$$

which is the same as the boxed expression above, concluding induction and proof. \square

14 March 8, 2019

We will continue with presentations today:

Problem 122

Prove that the major index (that is, the sum of the indices of the descents) is equidistributed with the number of inversions.

Solution by Yogeshwar Velinger. We're trying to show that

$$\sum_{w \in S_n} q^{\text{maj}(w)} = \sum_{w \in S_n} q^{\text{inv}(w)}.$$

The right hand side, as proved in class by induction, is

$$1(1+q)(1+q^2)\cdots(1+q+\cdots+q^{n-1}) = [n]_q!.$$

We'll show that the generating function for the major index is also this same polynomial. The base case $n = 1$ is clear, because there are no descents and no inversions. For the inductive step, suppose we have a permutation $w = (w_1, \dots, w_n)$. If we insert $n + 1$ into w , there are several cases:

- Case 1: if we insert $n + 1$ at the end, the major index stays the same.
- Case 2: Otherwise, we insert $n + 1$ somewhere in the middle. Appending 0 to the beginning of the sequence, (so that $w_0 = 0$), $n + 1$ must be inserted between w_i and w_{i+1} for some i .
 - Case 2a: If $w_i < w_{i+1}$ (ascent case), then inserting $(n + 1)$ adds a descent at spot $i + 1$ and then shifts all descents after it by 1 spot. So in this case, the major index increases by $i + 1 + d(i)$, where $d(i)$ is the number of descents in w at or after i .
 - Case 2b: If $w_i > w_{i+1}$ (descent case), then the descent moves from spot i to spot $i + 1$, and we also shift all later descents by 1. In this case, the major index just increases by $d(i)$.

Our goal is to show that the set of increments over all cases give us the integers 0 through n in some order. Notice that $d(i) + i + 1$ (in case 2a) is at most n , because the descents in $d(i)$ can occur at spots $i + 1, \dots, n - 1$, and also $d(i) < n$ in case 2b. So the $(n + 1)$ different increments all take on values between 0 and n inclusive, and we just need to show that they are all different.

Clearly $d(i) + i + 1$ is always positive, and an increase of $d(i)$ (from case 2b) only occurs if we already have an existing descent at i , so the case-2 insertion increments are all positive. It remains to check that we can't get the same number twice. We can't have $d(i) = d(j)$ for two different descent indices i, j , because one descent occurs before the other. Meanwhile, i and j are both ascent indices, then $d(i) + i + 1 = d(j) + j + 1 \implies d(i) - d(j) = j - i$. But that would mean we need descents at all spots between i and j , which is not possible since i is an ascent. Finally, suppose that i is a descent and j is an ascent. Then the only way the increments can be equal is if

$$d(i) = d(j) + j + 1 \implies d(i) - d(j) = j + 1,$$

but $d(i) - d(j)$ must always be less than j , so this cannot occur. This means that the increases in major index from inserting $(n + 1)$ are $\{0, 1, \dots, n\}$ in some order, so the generating function gets multiplied by $(q^0 + q^1 + \dots + q^n)$, completing the induction and proof. \square

Problem 123

Find the number of permutations $w \in S_n$ that are 231-avoiding and 4321-avoiding.

Solution by Congyue Deng. Denote the number of permutations that are both 231- and $k(k-1)\cdots 21$ -avoiding $T(n, k)$. We claim that

$$T(n+1, k) = \sum_{i=0}^n T(i, k)T(n-i, k-1).$$

To show this, write an arbitrary permutation $w \in S_{n+1}$ in one-line notation as $(\sigma, n+1, \tau)$, where σ contains i elements and τ contains the remaining $n-i$ elements. Since w is 231-avoiding, the elements in σ must all be smaller than the elements in τ (meaning that τ contains the integers from 1 through $n-i$, and σ contains the integers from $n+1-i$ through n). Other than that, σ and τ must each be 231 and $k(k-1)\cdots 1$ -avoiding, and in fact τ must be $(k-1)(k-2)\cdots 1$ -avoiding (because $(n+1)$ always contributes to a decreasing sequence), so we obtain the recurrence relation above by summing over the possible values of i .

We can now just repeatedly compute starting from small values of k . For $k=2$, we want a permutation which is 21-avoiding, and there is only one such permutation for all n (the identity permutation). Thus,

$$T(n+1, 3) = \sum_{i=0}^n T(i, 3)$$

with the initial conditions $T(0, 3) = T(1, 3) = 1, T(2, 3) = 2$, and by induction this implies that $T(n, 3) = 2^{n-1}$ for all $n \geq 1$ (and $T(0, 3) = 1$). Finally, we can plug in $k=4$ into our recurrence relation to obtain

$$T(n+1, 4) = 2^{n-1}T(0, 4) + 2^{n-2}T(1, 4) + \cdots + T(n-1, 4) + T(n, 4).$$

Subtracting the equations for $T(n+1, 4)$ and $T(n, 4)$, we find that this is equivalent to the linear recurrence

$$T(n+1, 4) = 3T(n, 4) - T(n-1, 4),$$

and plugging in the initial condition $T(1, 4) = 1, T(2, 4) = 2$ allows us to verify that our recurrence is satisfied by the alternating Fibonacci numbers $T(n, 4) = \boxed{F_{2n-1}}$, which is our desired answer. \square

We'll present a third solution of Problem 120 next:

Solution by Wanlin Li. Draw the semicircle arc-diagram corresponding to a set-partition; for example, $(1, 3, 5)$ in the same partition would correspond to semicircular arcs between 1 and 3 and between 3 and 5. If we have a set-partition of $[n]$ where i and $i+1$ are never in the same partition, this is equivalent to having no arcs of diameter 1. Now consider shrinking every arc by 0.5 in each direction (so that an arc with diameter i to j now has diameter $i+0.5$ to $j-0.5$); this creates an arc-diagram on the $(n-1)$ points $1.5, 2.5, \dots, n-0.5$, so it can be associated with a set-partition of $[n-1]$. Because this operation is bijective (inverse map expands each arc), we obtain a bijection between the two desired sets and thus they have equal size. The only detail to check is that shrinking and expanding indeed give us valid arc diagrams, but because the set of arcs with i as the original right endpoint is the same as the set of shrunk arcs with $i-0.5$ as the original right endpoint (and similar but with $i+0.5$ for left endpoints), we will never have two arcs that have the same right-endpoint or left-endpoint, which is the only constraint that we have. \square

Problem 124

Consider the biased drunk-walk. Find the probability that if the man starts at position i_0 and moves right with probability p at each step, he falls off after exactly m steps.

Solution by Sarah Wang. The drunk man needs to take a total of k steps to the right and $k + i_0$ steps to the left to end up falling off after m steps, so $k + (k + i_0) = m \implies k = \frac{m - i_0}{2}$. (We assume $m \geq i_0$ and that m, i_0 have the same parity, because otherwise the probability is zero.)

Draw the path that the drunk man takes as a Dyck path; we want to count the number of paths from $(0, i_0)$ to $(m - 1, 1)$ (and then go down to $(m, 0)$). Much like with the Catalan numbers, we can do a reflection argument and instead count the number of bad paths. All of the bad paths hit the line $y = 0$ in the xy -plane; pick the first point $(k, 0)$ where this happens, and reflect the path from $(k, 0)$ to $(m - 1, 1)$ over the y -axis to end at $(m - 1, -1)$ instead. This is a bijection, so the total number of bad paths is $\binom{m-1}{k-1}$, meaning the number of good paths is $\binom{m-1}{k} - \binom{m-1}{k-1}$. Each path occurs with probability $p^k(1-p)^{k+i_0}$, so our final answer is the probability of these two terms. \square

Problem 125

Prove the baby hook-length formula from Theorem 53, namely that the number of linear extensions of a poset whose Hasse diagram is a rooted tree is $\frac{n!}{\prod_{v \in T} h(v)}$.

Solution by Sanzeed Anwar. We proceed by strong induction – the base case $n = 1$ is easy to show. For the inductive step, consider a poset whose Hasse diagram is a rooted tree of n vertices. If we remove the root and its adjacent edges from the tree, we have a collection of subtrees T_1, T_2, \dots, T_k , with m_1, \dots, m_k vertices, where $m_1 + \dots + m_k = n - 1$. By strong induction, the number of linear extensions of T_i is just $\frac{m_i!}{\prod_{v \in T_i} h(v)}$ (removing the root does not change the hook length $h(v)$), so if we want a rooted tree on all n vertices, we can do this in

$$\frac{m_1! m_2! \cdots m_k!}{\prod_{v \neq \text{root}} h(v)} \binom{n-1}{m_1, m_2, \dots, m_k}$$

total ways, since we pick which m_1 nodes to use for T_1 , which m_2 nodes to use for T_2 , and so on, and then apply the baby hook-length formula on each T_i . Expanding the multinomial coefficient, this simplifies to

$$= \frac{(n-1)!}{\prod_{v \neq \text{root}} h(v)} = \frac{n!}{\prod_v h(v)},$$

as desired, finishing induction and proof. \square

Problem 126

Show that a permutation of length $mn+1$ has an increasing subsequence of length $m+1$ or decreasing subsequence of length $n+1$.

Solution (student did not identify themselves). For each element in the permutation, assign to it an ordered pair (i, j) , where i is the length of the longest increasing subsequence ending at that element and j is the length of the longest decreasing subsequence ending at that element. These ordered pairs are all distinct (if we have any two indices $a < b$, then either the longest increasing or longest decreasing subsequence ending at a can be extended by tacking on the element at b). Thus, we can't have $1 \leq i \leq m$ and $1 \leq j \leq n$ for all i, j by the pigeonhole principle (since there are only mn such pairs). \square

15 March 11, 2019

We have one more presentation today:

Problem 127

Let $k \leq \frac{n}{2}$. Find a bijection f from k -element subsets to $(n - k)$ -element subsets of $[n]$, satisfying $f(I) \supset I$ for any k -element subset I .

Solution by Chiu Yu Cheng. Note that the complement of a k -element subset has $n - k$ elements. Thus, it suffices to find a bijection $f' : I \rightarrow f'(I)$ from k -element subsets to k -element subsets, such that $f'(I) \cap I = \emptyset$ for all I .

Our function f' is constructed as follows: put the numbers $1, 2, \dots, n$ in a circle in order. Initialize $f'(I)$ to the empty set, and for each $x \in I$, move clockwise until we meet the first element not in I and $f'(I)$ already, and put that in $f'(I)$. Repeat this process until all elements of I have been accounted for.

We just need to show this is a bijection – because we’re mapping the set of k -element subsets to itself, it suffices to show that $f'(I)$ is well-defined regardless of the order in which we pick elements of I , and that an inverse map exists. To show the former, assign labels of $+1$ s to the elements of $[n]$ if they are in I and -1 s otherwise. Then an element $x \notin I$ is in $f'(I)$ if and only if there is a counterclockwise partial sum starting from $x - 1$ that is positive, and this is independent of the order of I chosen. Thus f' is well-defined. For the latter, if we are given $f'(I)$, we can recover the original I by going counterclockwise instead of clockwise and doing the same operation (initialize I to the empty set; for each element in $f'(I)$, move counterclockwise until we reach something not yet in I or $f'(I)$, and add it). Because the roles of 1 and -1 just swap, this map is also well-defined and gives us back our original set. \square

The above problem was a Google interview question a while ago. As we move into our next topics for this class, we’ll soon see another solution to this problem!

Recall the concepts of **posets** and **lattices** from a few lectures ago, in which we have a set L with operations \wedge (meet) and \vee (join). (See, for example, Fact 117.) We will now construct certain lattices, sometimes using the order relation \leq and sometimes using \wedge and \vee :

Definition 128

The **Boolean lattice** B_n contains the subsets of $\{1, \dots, n\}$, with the order relation $S_1 \leq S_2 \iff S_1 \subset S_2$.

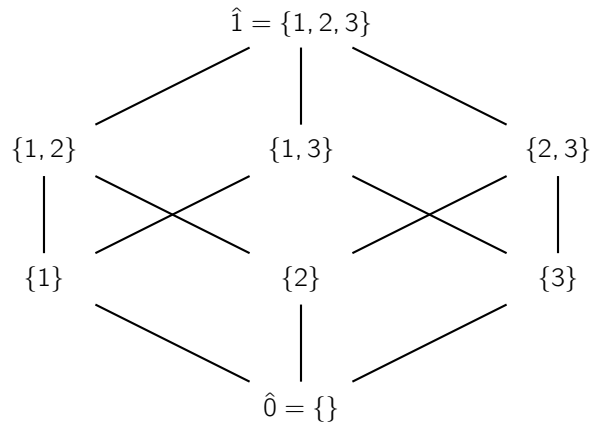
In particular, \wedge and \vee have nice interpretations in the Boolean lattice:

$$S_1 \wedge S_2 = S_1 \cap S_2, S_1 \vee S_2 = S_1 \cup S_2.$$

This may give some insight into why \wedge and \vee look the way they do!

Example 129

The Boolean lattice B_3 is shown below. Here, $\hat{0}$ denotes the smallest element of the lattice, while $\hat{1}$ denotes the largest element:



One observation we may have is that the \leq relations in this lattice trace out a 3-dimensional cube.

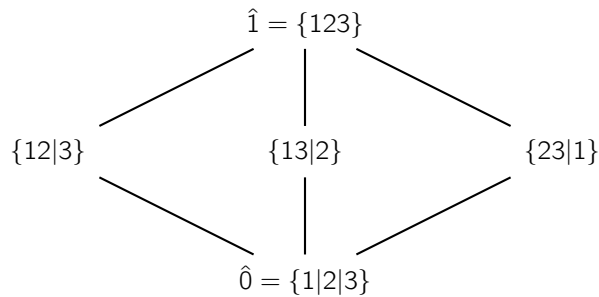
Definition 130

The **partition lattice** Π_n contains the set-partitions of $\{1, \dots, n\}$. They are ordered by **refinement**: in other words, $\pi \leq \sigma$ (equivalently, π **refines** σ or σ **coarsens** π) if each block of π is contained in some block of σ .

Here, the meet operation $\sigma \wedge \pi$ is the **common refinement** of σ and π , where we take all intersections of blocks. (For example, $(145|26|378) \wedge (146|2357|8) = (14|2|37|5|6|8)$.) On the other hand, the join operation $\sigma \vee \pi$ is the **finest common coarsening** of σ and π , but this is slightly harder to define. Basically, a sequence of elements a, a', a'', a''', \dots are all in the same block of $\sigma \vee \pi$ if a and a' are in the same block of π , a' and a'' are in the same block of σ , a'' and a''' are in the same block of π , and so on. (For example, $(145|26|378) \vee (146|2357|8) = (12345678)$.)

Example 131

The partition lattice Π_3 is shown below:



Definition 132

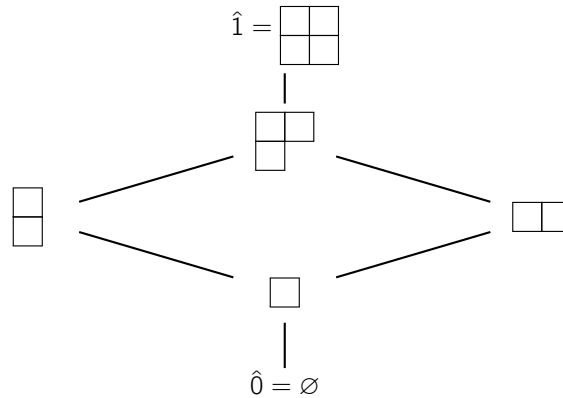
Young's lattice \mathbb{Y} is an infinite lattice of all Young diagrams ordered by containment (that is, $\lambda \leq \mu$ if the boxes of λ fit inside the boxes of μ).

The minimal element $\hat{0}$ of \mathbb{Y} is the empty Young diagram. From there, we can recursively build up the next layer of Young's lattice by adding a box to any of the Young diagrams in the current layer, so we have 1, 2, 3, 5 Young diagrams in the subsequent layers. Notice that for any given Young diagram, there will always be 1 more Young diagram connected above it than below it (if a Young diagram has c corners, then there are $(c + 1)$ places to add a box).

We can also define finite sublattices of \mathbb{Y} : for example, we can fix m, n , and define $L(m, n)$ to be the sublattice of Young diagrams that fit inside an $m \times n$ box (which we may recall are generated by the generating function $\begin{bmatrix} n \\ k \end{bmatrix}_q$).

Example 133

The lattice $L(2, 2)$ is a finite lattice with a unique maximal and minimal element. It is shown below:



Switching to the meet and join operations, the situation is very similar to the Boolean lattice: for any $\lambda, \mu \in \mathbb{Y}$, $\lambda \wedge \mu$ is the set-theoretic intersection, and $\lambda \vee \mu$ is the union. (We can indeed check that these are indeed always Young diagrams.)

Definition 134

A lattice (L, \wedge, \vee) is **distributive** if it satisfies the two distributive laws

- $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$,
- $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.

(Notice that multiplication and addition satisfy one of the two distributive laws but not the other one.)

Lemma 135

The Boolean lattice B_n is distributive, and so are Young's lattice \mathbb{Y} and its finite sublattices $L(m, n)$.

Proof. For B_n , this is just a set theoretic result: it is easy to check (by drawing a Venn diagram) that for any sets X, Y, Z , $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$, and similarly $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$. For \mathbb{Y} and $L(m, n)$, the meet and join operations are still unions and intersections (just boxwise), so the same proof works. \square

Example 136

However, not all lattices are distributive: the partition lattice Π_n for $n \geq 3$ is not a distributive lattice. For example, take $n = 3$, and let $x = (12|3)$, $y = (13|2)$, and $z = (23|1)$. Then

$$x \vee (y \wedge z) = x \neq \hat{1} = (x \wedge y) \vee (x \wedge z).$$

It turns out there is a very simple description of finite distributive lattices:

Definition 137

A subset I of a poset P is an **order ideal** if for any $x \in I$ and $y \leq x$, we also have $y \in I$.

In other words, order ideals are closed downward, so that they contain some “bottom part” of the poset’s Hasse diagram.

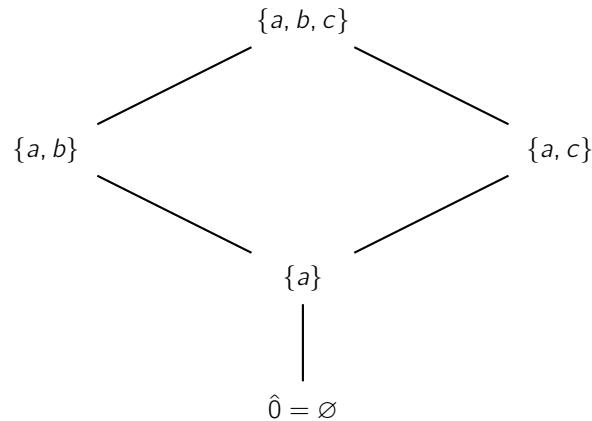
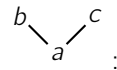
Definition 138

For any poset P , let $J(P)$ be the poset of all order ideals in P , ordered by containment.

Theorem 139 (Birkhoff’s fundamental theorem of finite distributive lattices)

The map $P \rightarrow J(P)$ is a one-to-one correspondence between finite posets and finite distributive lattices.

As an illustrative example, if our poset has elements a, b, c satisfying $a \leq b$ and $a \leq c$, then the order ideals are $\{\}, \{a\}, \{a, b\}, \{a, c\}$, and $\{a, b, c\}$. In other words, here’s the poset $J(P)$ corresponding to $P =$



We’ll look more into the proof of this result next time!

16 March 13, 2019

Recall from last lecture that if we have a poset P , then $J(P)$ is the poset of order ideals in P , ordered by inclusion.

Lemma 140

For any poset P , $J(P)$ is a distributive lattice.

Proof. To define a lattice, we need to define the meet and join operations. If we have two order ideals I and J , we set $I \wedge J = I \cap J$ and $I \vee J = I \cup J$ – it can be routinely checked that the intersection and union of two order ideals is also an order ideal. Since the usual union and intersection satisfy the distributive laws, $J(P)$ is indeed a distributive lattice. \square

The idea behind Birkhoff’s fundamental theorem of finite distributive lattices (Theorem 139) is that the converse is also true (specifically, every finite distributive lattice is of the form $J(P)$ for some P). We’ll go through the main idea of this result now:

Key idea of the proof. We already know how to get from $P \rightarrow J(P)$, so our goal is to reconstruct a poset from a finite distributive lattice L .

Definition 141

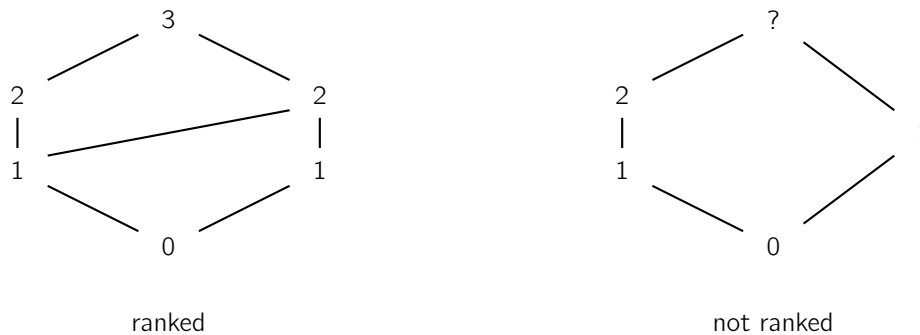
An element $x \in L$ is **join-irreducible** if it is not the minimal element of L (which always exists by repeatedly applying \wedge) and we cannot express it as $x = y \vee z$ for $y, z < x$.

For example, in the lattice $J(P)$ in the diagram above, all elements except \emptyset and $\{a, b, c\}$ join-irreducible. Notice that if we remove those elements, we get a poset isomorphic to the original P – it turns out in general, we construct P to be the **subset of L of all join-irreducible elements!** Showing that these are isomorphic is left as an exercise to us. □

Definition 142

A poset P is **ranked** if there is a function $\rho : P \rightarrow \{0, 1, 2, \dots\}$, such that $\rho(x) = 0$ for any minimal element x of P and $\rho(y) = \rho(x) + 1$ if $x < y$.

So far, all the posets that we've been discussing have been ranked, with the idea being that all elements live on different "levels." In the diagram (with numbers representing values of ρ), the left poset is ranked, but the right poset is not.



Proposition 143

Any finite distributive lattice is ranked and has the **modularity property** $\rho(x \vee y) + \rho(x \wedge y) = \rho(x) + \rho(y)$.

Proof. Write our finite distributive lattice L as $J(P)$. Then for any arbitrary element $I \in L$, which we can view as an order ideal of P , we can assign it a rank of $\rho(I) = |I|$. To check that this is the case, we just need to show that if $I < J$, they only differ in one element: this is true because $I < J$ means I is strictly contained in J , and we can always remove one "upper" element from a nonempty order ideal and still have an order ideal. The result then follows from the fact that $|A \cup B| + |A \cap B| = |A| + |B|$ for any sets A, B . □

Definition 144

Let P be a finite ranked poset. The **rank numbers** r_i of P are the number of elements in P of rank i . These rank numbers form a vector (r_0, r_1, \dots, r_N) , where N is the maximal rank of any element in P . Then P is **rank-symmetric** if $r_i = r_{N-i}$ for all i , and it is **unimodal** if $r_0 \leq r_1 \leq \dots \leq r_j \geq r_{j+1} \geq \dots \geq r_N$ for some j .

For example, in our most recent example of a ranked poset, we have rank numbers $(1, 2, 2, 1)$, so the poset is both unimodal and rank-symmetric.

Definition 145

Let P and Q be two posets. The **product poset** $P \times Q$ is the poset whose elements are pairs $(p, q), p \in P, q \in Q$, with order relation

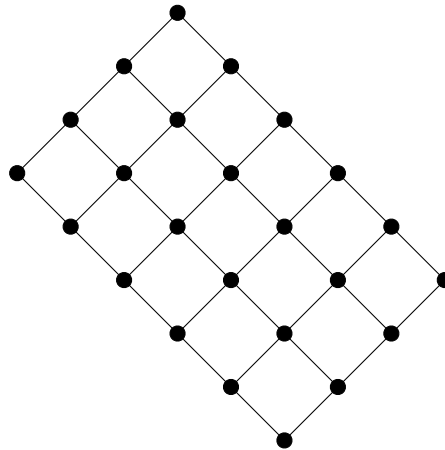
$$(p, q) \leq (p', q') \iff p \leq p' \text{ and } q \leq q'.$$

Notice that even if all pairs of elements in P and Q are comparable, this does not need to be the case in $P \times Q$.

Example 146

Let $[n]$ denote the poset whose Hasse diagram is an increasing chain of n elements. (This is clearly rank-symmetric and unimodal.) Then $[m] \times [n]$ is a 45-degree-rotated grid, where the element (x, y) has rank $x + y$.

Below is an illustration of the poset $[4] \times [6]$, where each horizontal level consists of elements of the same rank:



In general, the rank numbers for $[m] \times [n]$ are $(1, 2, \dots, k - 1, k, \dots, k, k - 1, \dots, 2, 1)$, where $k = \min(m, n)$. Thus this poset is also rank-symmetric and unimodal.

We now want to see what happens if we apply the function J to these simple posets. $J([n])$ is just $[n + 1]$ (the order ideals are the n chains containing the bottom element, plus the empty ideal), but $J([m] \times [n])$ is more interesting. The idea is that **order ideals correspond to Young diagrams** if we replace elements with boxes, so the order ideals are exactly the Young diagrams that fit inside our grid:

$$J([m] \times [n]) = L(m, n)$$

is the poset of Young diagrams that fit inside an $m \times n$ rectangle. Thus, the rank numbers r_i of $J([m] \times [n])$ are the counts of Young diagrams with a fixed number of squares, and thus these are the Gaussian coefficients which show up like in Theorem 57:

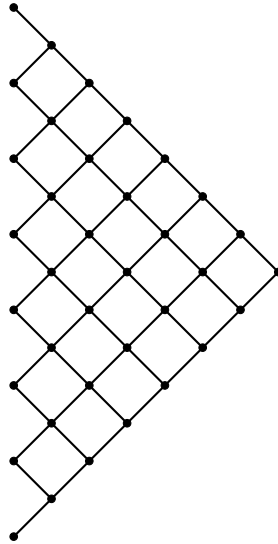
$$\begin{bmatrix} m + n \\ n \end{bmatrix}_q = r_0 + r_1 q + \dots + r_{mn} q^{mn}.$$

After some more work (which we will do **later in the course**), we arrive at the following result:

Theorem 147

$L(m, n)$ is rank-symmetric and unimodal.

Looking at a special case ($m = 2$ will provide even more insight). If we want to visualize $J([2] \times [n])$, then we want to draw the poset of Young diagrams that fit inside a $2 \times n$ rectangle. The corresponding Hasse diagram looks like a triangle:



And now if we look at the order ideals of this triangular Hasse diagram (so we're trying to analyze $J(J([2] \times [n]))$), we can observe that the order ideals here correspond to **shifted Young diagrams**! So $J(J([2] \times [n]))$ is the poset of shifted Young diagrams (ordered by inclusion) which fit inside a shifted Young diagram with $n, n - 1, \dots, 1$ boxes in the first n rows. Showing that this is also rank-symmetric and unimodular is left as an exercise to us.

Theorem 148 (Sperner (1928))

Let S_1, \dots, S_M be different subsets of $\{1, 2, \dots, n\}$, such that for all $i, j, S_i \not\subseteq S_j$. Then $M \leq \binom{n}{n/2}$.

To understand why this is related to posets, we'll need to look at multiple elements of our posets at once:

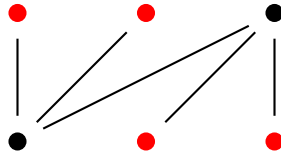
Definition 149

A **chain** C in a poset P is a set of elements such that any two elements are comparable (for any $a, b \in P$, either $a \leq b$ or $b \leq a$), meaning that we have a total ordering on C . On the other hand, an **antichain** A in P is a set where no two elements are comparable.

Definition 150

Let P be a finite ranked poset with rank numbers r_0, r_1, \dots, r_N . Then P is **Sperner** if the maximal size M of any antichain in P satisfies $M = \max(r_0, \dots, r_N)$.

Note that we always have $M \geq \max(r_0, r_1, \dots, r_N)$, because we can always take all of the elements of a given rank and they will not be comparable. A Sperner poset is one where equality holds, and Sperner's theorem really says that **the Boolean lattice is Sperner**, because the central binomial coefficient is the maximal rank number of the Boolean lattice! We'll do the proof next time, but for reference, below is an example Hasse diagram of a **non-Sperner poset**, because the rank numbers are 3, 3 but there is an antichain of length 4.



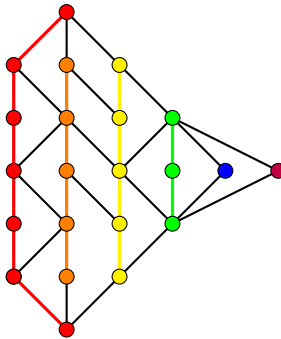
17 March 15, 2019

Last time, we discussed ranked posets, studying properties of their rank numbers r_i . In particular, a ranked poset P is **rank-symmetric** if $r_i = r_{N-i}$ for all i , **unimodal** if the rank numbers increase and then decrease, and **Sperner** if the maximal size of an antichain is the maximal rank number. It turns out that there's a property of posets that implies all three properties, which we'll work towards now.

Definition 151

Let C be a chain of a ranked poset P . Then C is **saturated** if its elements are $\{x_0 < x_1 < x_2 \cdots < x_m\}$ (in particular, $\rho(x_i) = \rho(x_0) + i$ for all i). A **symmetric chain decomposition** is a decomposition of P 's elements into a disjoint union of saturated chains C_i , such that for all chains $C_i = \{x_0 < \cdots < x_{\ell_i}\}$, we have $\rho(x_{\ell_i}) + \rho(x_0) = N$.

For example, below is a poset with its symmetric chain decomposition:



Lemma 152

If P has a symmetric chain decomposition, then it is rank-symmetric, unimodal, and Sperner.

Proof. Each chain contributes 1 to some set of rank numbers which is symmetric about $\frac{N}{2}$, so the rank numbers will be symmetric (because the sum of palindromic vectors is palindromic). Unimodality is also clear, because the sum of vectors that are each unimodal and symmetric about a fixed mean is also unimodal.

To show that P is Sperner, note that the middle rank number $\lfloor \frac{N}{2} \rfloor$ (also true for $\lceil \frac{N}{2} \rceil$) is exactly the number of symmetric chains we have in our symmetric chain decomposition. After all, we can write $P = C_1 \cup C_2 \cup \cdots \cup C_m$, and each chain intersects the middle level exactly once (because each C_i is saturated and symmetric about the middle). By unimodality, this middle rank number is maximal, and any antichain cannot contain two elements in the same chain C_i . Thus, because there are m chains, any antichain can only contain at most m elements, as desired. \square

To understand why this is relevant, we'll now work towards the proof of Theorem 148.

Proposition 153

Let $[n]$ denote the poset with n elements in a single chain. Then the Boolean lattice can be written as $B_n = [2] \times [2] \times \cdots \times [2]$ (with n terms in the product).

For example, $[2] \times [2]$ is a square, $[2] \times [2] \times [2]$ is a 1-skeleton of a cube, and so on. (This can be shown by induction, having each $[2]$ denote whether a given element of $\{1, 2, \dots, n\}$ is included or not.)

Theorem 154 (de Bruijn (1948) and generalization)

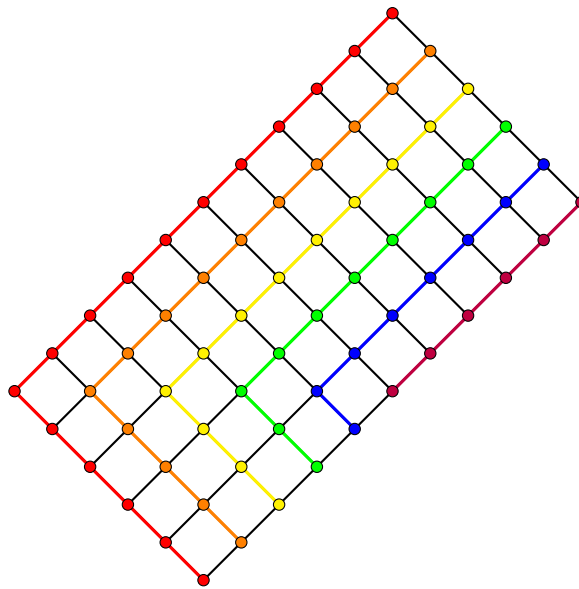
B_n has a symmetric chain decomposition. More generally, for any integers a_1, a_2, \dots, a_n , any product $[a_1] \times [a_2] \times \dots \times [a_n]$ has a symmetric chain decomposition.

We can show this result by combining two lemmas:

Lemma 155

For any a, b , $[a] \times [b]$ has a symmetric chain decomposition.

We'll do a proof by picture:



Lemma 156

If posets P and Q have symmetric chain decompositions, then $P \times Q$ has a symmetric chain decomposition.

Proof. Write out the symmetric chain decompositions as $P = C_1 \cup C_2 \cup \dots \cup C_k$ and $Q = C'_1 \cup \dots \cup C'_l$. Then we can write P as a disjoint union of products of chains

$$P \times Q = \bigcup_{(i,j)} C_i \times C'_j.$$

But each saturated chain $C_i \times C'_j$ is of the form $[a] \times [b]$, so we can pick a symmetric chain decomposition for each $C_i \times C'_j$ as we did in Lemma 155. This gives a symmetric chain decomposition for the whole poset, as desired. \square

With that, we've now proved Sperner's theorem, and let's now go back to looking at finite posets in general. Given any poset P , recall that we defined $M(P)$ to be the maximum number of elements in any antichain of P .

Theorem 157 (Dilworth (1950))

Let $m(P)$ be the minimum number of disjoint chains needed to cover all elements of P . Then for any finite poset P , $M(P) = m(P)$.

There's also a dual version of this theorem:

Theorem 158 (Minsky (1971))

If we flip the words "chain" and "antichain" (in the definition of $M(P)$ and $m(P)$) in Dilworth's theorem, then the statement still holds.

In fact, this duality manifests even more generally in a way that will connect to previous material:

Definition 159

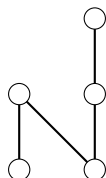
Given a poset P , let ℓ_k be the maximum number of elements in a union of k (not necessarily disjoint, not necessarily saturated) antichains in P . Similarly, define m_k to be the maximum size of a union of k chains in P .

For example, ℓ_1 is $M(P)$, the maximum number of elements in an anti-chain.

Theorem 160 (Greene (1976))

Define $\lambda(P) = (\ell_1, \ell_2 - \ell_1, \dots)$ and $\mu(P) = (m_1, m_2 - m_1, \dots)$. Then λ and μ are both partitions of n that are weakly decreasing, and they are **conjugates** (that is, their Young diagrams are transposes of each other).

For example, consider the following poset:



Here, we can cover at most 2, 4, and 5 elements with 1, 2, and 3 antichains, so

$$\lambda = (2, 4 - 2, 5 - 4) = (2, 2, 1) = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \\ \hline \end{array}.$$

Meanwhile, we can cover 3, 5 elements with 1, 2 chains, so

$$\mu = (3, 5 - 3) = (3, 2) = \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \\ \hline \end{array}.$$

In this language, Dilworth's theorem says that the first row of λ has the same number of boxes as the first column of μ , and Minsky says the same thing with row and column swapped. But recall the Schensted correspondence (and the previous result by Greene, which we discussed in Theorem 41), which explained that the shape λ of a Young diagram tells us about increasing subsequences in permutations. That previous result can be explained by the fact that we can make a poset out of a permutation, such that increasing subsequences are chains and decreasing subsequences are antichains! We'll see this in action next time.

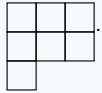
18 March 18, 2019

Last time, we discussed a duality between chains and antichains of a poset, in which the Young diagrams formed were transposes of each other. Prior to that, we had discussed the Schensted correspondence: every permutation $w \in S_n$ could be associated to a pair (P, Q) of SYTs of shape λ , in which the rows and columns encode information about increasing and decreasing subsequences. We'll start today by showing the relationship between these ideas. Given any $w \in S_n$, we can construct a poset P with elements $\{1, 2, \dots, n\}$, with relation \leq_p dictated by

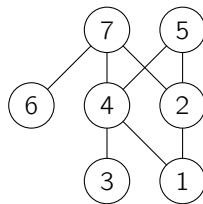
$$i <_p j \iff i < j \text{ and } w_i < w_j.$$

Example 161

Let's take $w = (3, 5, 2, 4, 7, 1, 6)$, for which the Schensted correspondence gives us a Young diagram of shape



The Hasse diagram for the associated poset is shown below:



Notice that in any such permutation, increasing subsequences correspond to chains (because the increasing numbers must follow each other), and decreasing subsequences correspond to antichains (because no smaller number comes before a larger number)! So Greene's theorem gives chain numbers of $(3, 6 - 3, 7 - 1) = (3, 3, 1)$, and antichain numbers of $(3, 5 - 3, 7 - 5) = (3, 2, 2)$, as we expect.

We'll now turn back to other properties that we analyzed earlier in the class and look at them with a new perspective. Recall that we used the Schensted correspondence to prove the identity

$$\sum_{\lambda: |\lambda|=n} (f^\lambda)^2 = n!,$$

where f^λ is the number of standard Young tableaux of a shape λ . Today, we're going to do a simpler proof of this result that is more general. Consider Young's lattice \mathbb{Y} , which is isomorphic to $J(\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0})$, the lattice of order ideals in a quadrant of \mathbb{Z}^2 . Let \mathbb{Y}_n be the set of all Young diagrams with exactly n boxes (which is the n th level of the lattice). Then we can define the **partition number**

$$p(n) = |\mathbb{Y}_n|,$$

which is the number of ways to write n as a sum (disregarding order).

Proposition 162

A standard Young tableau of shape λ corresponds to a path (that is, a saturated chain) in the Hasse diagram of \mathbb{Y} from \emptyset to λ .

(To see this, notice that we add a box to our Young diagram each time we traverse an edge in our Hasse diagram. So for a given path, we can place a 1 in the first box added, then place a 2 in the next one tacked on, repeating until we get to λ .) In the Frobenius-Young identity and also in the Schensted correspondence, we have **pairs** of Young tableaux, which correspond to having two paths from the empty Young diagram to λ . So if we reverse the second path, we can think of (P, Q) as a **path in \mathbb{Y} from the empty shape back to itself with n up steps followed by n down steps**. Our goal is then to count the number of ways to draw this path!

To formalize this algebraically, let $\mathbb{R}[\mathbb{Y}_n]$ be the linear space of formal linear combinations of Young diagrams with n boxes, containing elements that look like

$$\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \\ \hline \end{array} + e \cdot \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} - \pi \cdot \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}.$$

So $\mathbb{R}[\mathbb{Y}_n]$ is isomorphic to $\mathbb{R}^{\rho(n)}$, and $\mathbb{R}[\mathbb{Y}] = \bigoplus_{n \geq 0} \mathbb{R}[\mathbb{Y}_n]$. Next, define **up** and **down** operators: the up operator U acts on Young diagrams via

$$U = \lambda \rightarrow \sum_{\mu: \lambda < \mu} \mu$$

(in other words, we take λ to the sum of all Young diagrams with one box appended to λ), and similarly, the down operator D acts on Young diagrams via

$$\mu \rightarrow \sum_{\lambda: \lambda < \mu} \lambda.$$

For example,

$$U \left(\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \\ \hline \end{array} + \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}$$

and

$$D \left(\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}.$$

We can now rephrase our question: the number of pairs (P, Q) of Young tableaux of n boxes is the coefficient of \emptyset (the bottom element in the poset – not zero!) in the expression $D^n U^n \emptyset$, and we wish to show that this is $n!$. We need the following key identity:

Proposition 163

Let U and D be the up and down operators defined above. If I is the identity operator, then

$$[D, U] = DU - UD = I.$$

Proof. Since D and U remove and add a box to λ , respectively, all nonzero terms in $(DU - UD)\lambda$ will involve Young diagrams with the same number of boxes as λ . Consider the coefficient of μ in $(DU - UD)\lambda$ – first of all, if $\lambda \neq \mu$, and we want to get from λ to μ using the operations D and U once, we must add a box a and remove a box r , and $a \neq r$ because $\lambda \neq \mu$. But these operations can always be done in either order, so any contribution to μ would cancel out between the DU and UD terms.

However, if $\lambda = \mu$, then the contribution to $(DU - UD)\lambda$ comes from adding a box and then removing that same box, or first removing a box and then adding that same box back. The number of boxes we can remove is the number of inner corners, and the number of boxes we can add is the number of outer corners. Since these always differ by 1,

we get a coefficient of 1, and $(DU - UD)\lambda = \lambda$, meaning $DU - UD = I$, as desired. \square

This next concept was introduced by Stanley:

Definition 164

A **differential poset** is a ranked (infinite) poset with a unique minimally ranked element $\hat{0}$, such that if we define U and D in the same way as for \mathbb{Y} (where U of an element is the sum of the elements directly above it, and D of that element is the sum of the elements directly below it), we have $[U, D] = I$.

Combinatorially, this means that for any $x \neq y$ on the same level of a differential poset, there are the same number of ways to go up and then down, or down and then up, from x to y :

$$\#\{u : u \succ x \text{ and } u \succ y\} = \#\{v : v \prec x \text{ and } v \prec y\}$$

On the other hand, for any element x , there is one more “upper neighbor” than “lower neighbor” of x in the Hasse diagram:

$$\#\{u : u \succ x\} = 1 + \#\{v : v \prec x\}$$

Fact 165

Differential posets get their name from the related operators u multiplying a polynomial $f(x)$ by x , and d taking its derivative:

$$(du - ud)f(x) = (xf(x))' - xf'(x) = f(x).$$

It turns out that this structure of a differential poset is actually what is needed to prove the Frobenius-Young identity:

Theorem 166 (Stanley)

For any differential poset and any $n \in \mathbb{Z}_{\geq 0}$, we have

$$D^n U^n \hat{0} = n! \hat{0}.$$

Proof. We use the following lemma, which can be proved by just algebraic manipulation:

Lemma 167

For any $n \in \mathbb{Z}_{\geq 0}$, we have $DU^n = nU^{n-1} + U^n D$.

Proof. Induct on n ; the base cases $n = 0, 1$ are clear. For the inductive step, notice that

$$DU^{n+1} = (DU^n)U = (nU^{n-1} + U^n D)U = nU^n + U^n(UD + I) = (n + 1)U^n + U^{n+1}D,$$

which is the desired result. \square

Our theorem then follows by induction on n . The base case $n = 0$ is clear, and for the inductive step, we apply the lemma:

$$D^n U^n(\hat{0}) = D^{n-1}(DU^n)\hat{0} = D^{n-1}(nU^{n-1} + U^n D)\hat{0},$$

and now the second term contributes nothing because $D\hat{0} = 0$. So we're just left with

$$D^{n-1}nU^{n-1}\hat{0} = nD^{n-1}U^{n-1}\hat{0} = n(n-1)\hat{0} = n!\hat{0}$$

by the inductive hypothesis, completing induction and proof. □

Next time, we'll think about other differential posets and their combinatorial interpretations!

19 March 20, 2019

Recall that a **differential poset** is a ranked poset with a minimal element $\hat{0}$, where we define **up** and **down** operators such that $[D, U] = DU - UD = I$, where I is the identity operator. Combinatorially, this is equivalent to the following two conditions:

- For any $x \neq y$ in the same level, if there are a elements that cover both x and y , there are a elements covered by x and y .
- For any x , there is one more element covering x than elements covered by x .

We found an example last time – Young's lattice is a differential poset. It's natural to ask whether there are any other differential posets, and we'll take a constructive approach here. We must start with a unique minimal element $\hat{0}$, and there must be exactly 1 element above $\hat{0}$, and then 2 elements above that. Now, for the two elements x and y we just constructed, we need one element to cover both x and y , and then each of x and y need to have an additional edge to another element.

More generally, here's how we can construct the $(n+1)$ th level of a differential poset: to satisfy the first condition, reflect the previous level's edges up. In other words, take the bipartite graph between the n th and $(n-1)$ th levels, and flip it over the n th level to create new vertices on the $(n+1)$ th level. Now, each vertex needs an extra edge, so we will connect each vertex on the n th level to a (different) additional vertex on the $(n+1)$ th level. (This is not the only way to construct a differential poset's next level, because we don't necessarily need to do the "reflection" procedure, but it does give us an interesting object.)

The rank numbers of this differential poset are 1, 1, 2, 3, 5, 8, ... (the Fibonacci numbers), because when we construct the $(n+1)$ th level, we add one vertex for every vertex in the $(n-1)$ th level, as well as one vertex for every vertex in the n th level.

Theorem 168

The process above gives a valid lattice (known as the **Fibonacci lattice**). In other words, any two elements of the Fibonacci lattice have a well-defined meet and join.

(The proof of this result can be seen by assigning each element a sequence of 1s and 2s, where 1 and 2 correspond to adding from the n th and $(n-1)$ th layer, respectively.) Comparing this lattice to Young's lattice, in which the rank numbers start off 1, 1, 2, 3, 5, 7, we see that the rank numbers (and structures) will be different. However, the two lattices have very similar properties – for example, if we find the number of walks from $\hat{0}$ that go up n steps and then down n steps for the Fibonacci lattice, it will be $n!$ as well (since we only need the identity $DU - UD = I$).

We'll take a second look at where this $n!$ term is actually coming from, though – consider the expression $D^n U^n \hat{0}$, written out as

$$DD \dots DDUU \dots UU\hat{0}.$$

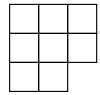
Think of the operators D and U as **particles** and **antiparticles**, where any D and U can collide (and cancel out) or go past each other, because $DU = UD + I$. But each D must collide with one of the U s before reaching the $\hat{0}$, because $D\hat{0} = 0$. So each D is matched with a U , and that means we really have a permutation of n D s.

More generally, we can pick any word with n U 's and n D 's, and we'll get an identity that looks something like

$$DDUDUU\hat{0} = c\hat{0}.$$

The number c is then the number of paths that go (up, up, down, up, down, down) in a differential poset (in this case it's 4, and we are trying to find a more systematic way of evaluating it). Basically, we have to find the total number of ways to match all D s with U s, having each D matches with a U to its right.

The key insight is that we can describe this setup with a rook placement! First, we trace out a path from the bottom left to top right of an $n \times n$ board, going right when we see a D and going up when we see a U , and draw the Young diagram in the upper left corner. For example, $DDUDUU$ gives the shape



Then any rook placement gives us a valid pairing of the corresponding D s and U s! Specifically, we have a Young diagram of form $\nu = (\nu_1, \dots, \nu_n)$, where the number of boxes in the i th row is the number of D s that appear before the $n + 1 - i$ th U in our sequence. Each rook then corresponds to a pairing between a U and a D (the i th row, j th column corresponds to pairing the $(n + 1 - i)$ th D with the j th U).

Theorem 169

Let W be a word of n D s and n U s, corresponding to a Young diagram $\nu = (\nu_1, \dots, \nu_n)$. Then the constant c in the equation $W\hat{0} = c\hat{0}$ is the number of placement of n non-attacking rooks in the shape ν , which is

$$\nu_n(\nu_{n-1} - 1)(\nu_{n-2} - 2) \cdots (\nu_1 - n + 1).$$

Proof. We've already explained the combinatorial connection between rook placements and the coefficient c ; to get the formula, we add rooks from bottom to top. For the first rook, we have ν_n choices, so the second rook has all ν_{n-1} choices available except the column that the first rook is in, and in general the i th rook has all of its ν_{n+1-i} choices available except the $(i - 1)$ columns that are already taken. Multiplying these terms gives us the result. \square

With that, we'll turn our attention back to the **unimodality of Gaussian coefficients** (which we stated but never proved). Recall that the Gaussian coefficients a_n come up in the expression

$$\begin{bmatrix} k + \ell \\ \ell \end{bmatrix}_q = \sum_{n=0}^{k\ell} a_n q^n,$$

where a_n is the number of Young diagrams that fit within a $k \times \ell$ box. We've already shown previously that these coefficients are symmetric (by taking the complement), and the result of unimodality was first formulated by Cayley in 1856 and proved by Sylvester in 1878:

Theorem 170 (Unimodality of the Gaussian coefficients)

The Gaussian coefficients in the equation $\begin{bmatrix} k + \ell \\ \ell \end{bmatrix}_q = \sum_{n=0}^{k\ell} a_n q^n$ satisfy $a_0 \leq a_1 \leq \dots \leq a_{\lfloor \frac{k\ell}{2} \rfloor} \geq \dots \geq a_{k\ell}$.

Proof. We'll state our problem in terms of Young diagrams: let P be the product poset $[k] \times [\ell]$, which we'll identify with a $k \times \ell$ rectangle. As we've previously discussed, the lattice of order ideals of P is the lattice of Young diagrams that fit inside a k by ℓ rectangle:

$$J(P) \sim L(k, \ell).$$

Let $J(P)_n$ to be the set of all Young diagrams in $J(P)$ on the n th level (containing n boxes). It suffices to show (by symmetry of the coefficients) that $|J(P)_{n+1}| \geq |J(P)_n|$ for all $n < \frac{k\ell}{2}$.

For a given Young diagram $\lambda \in J(P)_n$, let $\text{Add}(\lambda)$ denote the set of all boxes $x \in P$ such that $\lambda \cup \{x\} \succ \lambda$. Similarly, define $\text{Remove}(\lambda)$ to be the set of boxes $y \in P$ such that $\lambda \setminus \{y\} \prec \lambda$. Recall that in Young's lattice, the number of addable boxes is always one more than the number of removable boxes, but this is no longer true in our setting because we can't add boxes outside of our $k \times \ell$ rectangle.

Lemma 171

Fix $P = [k] \times [\ell]$ and $n < \frac{k\ell}{2}$. Suppose there is a **weight function** $w : P \rightarrow \mathbb{R}_{>0}$, such that for any $\lambda \in J(P)_n$, we have

$$\sum_{x \in \text{Add}(\lambda)} w(x) > \sum_{y \in \text{Remove}(\lambda)} w(y).$$

Then the number of elements in $J(P)_{n+1}$ is at least the number of elements in $J(P)_n$.

Here's an example of a weight function for $P = [3] \times [4]$ (we'll explain where it comes from later):

12	12	10	6
10	12	12	10
6	10	12	12

We can check that for any Young diagram with $n < 6$ boxes, the set of addable boxes has total weight larger than the set of removable boxes. To prove this lemma, we'll introduce the **weighted** up and down operators

$$U : \lambda \rightarrow \sum_{\substack{\mu = \lambda \cup \{x\}, \\ x \in \text{Add}(\lambda)}} \sqrt{w(x)} \cdot \mu, \quad D : \lambda \rightarrow \sum_{\substack{\mu = \lambda \setminus \{y\}, \\ y \in \text{Remove}(\lambda)}} \sqrt{w(y)} \cdot \mu.$$

We can think of these as linear operators on $\mathbb{R}^{J(P)}$, and under the natural basis (of Young diagrams), it is clear that $D = U^T$ are transpose matrices (with nonzero entries equal to $\sqrt{w(x)}$ for various boxes x).

Lemma 172

$H = [D, U] = DU - UD$ is a diagonal matrix, and for any Young diagram $\lambda \in J(P)$, we have

$$H\lambda = \left(\sum_{x \in \text{Add}(\lambda)} w(x) - \sum_{y \in \text{Remove}(\lambda)} w(y) \right) \lambda.$$

Proof. The matrix H is diagonal because (just like in the unweighted case) adding a box and removing another box can be done in either order, so the contributions cancel out in $DU - UD$. On the other hand, to compute the diagonal entries of $DU - UD$, we can add any box and remove it, each of which gives $\sqrt{w(x)}^2 = w(x)$, or remove any box and add it back, each of which contributes $-w(y)$. □

Furthermore, because D and U are transposes of each other, we can write

$$DU = UD + H = UU^T + H.$$

Then UU^T is positive semi-definite, and H is a positive definite matrix (because it is diagonal and the diagonal entries are positive by Lemma 171), so this means DU must be positive definite as well, meaning that it has a positive determinant. We'll continue this argument next time! \square

20 March 22, 2019

We'll start today by finishing the proof of unimodality of the Gaussian coefficients – despite the setting in which the problem was introduced, it's easier to use linear algebra instead of a combinatorial proof.

Proof, continued. We'll do a quick review and set up our notation more carefully today: Gaussian coefficients are the coefficients of the q -binomial coefficient $\begin{bmatrix} k + \ell \\ k \end{bmatrix}_q$, and we wish to show that they are increasing and then decreasing. The idea is to consider the linear space V_n of formal linear combinations of Young diagrams λ in a k by ℓ rectangle with n squares and to prove that $\dim V_n \leq \dim V_{n+1}$ for $n < \frac{k\ell}{2}$. We do this by defining a weight function $w : P \rightarrow \mathbb{R}_{>0}$ and then defining the **weighted up and down operators** $U_n : V_n \rightarrow V_{n+1}, D_n : V_{n+1} \rightarrow V_n$

$$U_n \lambda = \sum_{\substack{\mu = \lambda \cup \{x\} \\ \mu \triangleright \lambda}} \sqrt{w(x)} \mu, \quad D_n \lambda = \sum_{\substack{\mu = \lambda \setminus \{x\} \\ \mu \triangleleft \lambda}} \sqrt{w(x)} \mu.$$

(Here, $\mu = \lambda \cup \{x\}, \mu \triangleright \lambda$ is equivalent to $x \in \text{Add}(\lambda)$, and similar for $\text{Remove}(\lambda)$.) Below is a diagram, where λ is highlighted in yellow, the As are part of $\text{Add}(\lambda)$, and the Rs are part of $\text{Remove}(\lambda)$:

		R	A
	A		
R			

The commutator $H_n = D_n U_n - U_{n-1} D_{n-1}$ then takes any element in V_n to another element in V_n . We can represent U_n with an a_{n+1} by a_n matrix, and we can represent $D_n = U_n^T$ (because all nonzero entries in U_n have \sqrt{x} in the entry (a, b) , where a and b differ by the box x , and entries in D_n have \sqrt{x} in the entry (b, a)). Last time, we showed that H_n is diagonal and has entries given by Lemma 172.

So now let's assume we can find a weight function w so that the matrix H_n has positive diagonal entries, meaning the eigenvalues are all positive. Then

$$D_n U_n = U_{n-1} D_{n-1} + H_n = U_{n-1} U_{n-1}^T + H_n.$$

$U_{n-1} U_{n-1}^T$ is positive semi-definite (since for any matrix A , $x A A^T x$ is the square of the standard dot product of $A^T x$ with itself), and H_n is positive definite, so their sum is positive definite (this is a fact from linear algebra!) This means $D_n U_n$ has nonzero determinant, and therefore the rank of $D_n U_n$ is a_n for any n . We can finish with the following fact from linear algebra:

Fact 173

Let A be an $m \times n$ matrix and let B be an $n \times k$ matrix. Then $\text{rank}(AB) \leq \min(m, n, k)$ (because the rank of a matrix is always smaller than both dimensions, and $\text{rank}(AB) \leq \text{rank}(A), \text{rank}(B)$).

In particular, the rank of $D_n U_n$ is a_n , but D_n is an $a_n \times a_{n+1}$ matrix and U_n is $a_{n+1} \times a_n$, so

$$a_n \leq \min(a_n, a_{n+1}) \implies a_n \leq a_{n+1}.$$

which is what we have been aiming to show! So it suffices to find a weight function w , and we can now explain how we constructed the one last lecture. One such function $w : [k] \times [\ell] \rightarrow \mathbb{R}_{>0}$ that works is

$$w(i, j) = (i - j + \ell)(j - i + k), 1 \leq i \leq k, 1 \leq j \leq \ell.$$

For example, the weight function for $(k, \ell) = (3, 4)$ that we saw last lecture,

12	12	10	6
10	12	12	10
6	10	12	12

has all weights of the form $n(7 - n)$ for some positive integer n , and it has larger values closer to the center diagonal. To see why this weight function is valid, we can prove the following result:

Lemma 174

For all λ contained in a $k \times \ell$ box,

$$w_\lambda = \sum_{x \in \text{Add}(\lambda)} w(x) - \sum_{y \in \text{Remove}(\lambda)} w(y) = k\ell - 2|\lambda|,$$

which is positive as long as $n = |\lambda| < \frac{k\ell}{2}$.

This final part of the proof is left as an exercise to us! □

Let's shift gears now and talk more about **partitions** – recall that $p(n)$ is the number of partitions of n , which is also the number of Young diagrams with n boxes.

Proposition 175

The generating function for $p(n)$ can be written as

$$\sum_{n \geq 0} p(n)q^n = \frac{1}{(1 - q)(1 - q^2)(1 - q^3)\dots}$$

Expanding out each term on the right-hand side as $\frac{1}{1 - q^k} = 1 + q^k + q^{2k} + \dots$, we can compute the partition number $p(n)$ as the coefficient of q^n in

$$(1 + q + q^2 + \dots)(1 + q^2 + q^4 + \dots)(1 + q^n + \dots).$$

In particular, it's okay to only consider the first n terms, since all other terms will either contribute 1 or a power of q larger than n .

A handwavy proof. We know that

$$\left[\begin{matrix} k + l \\ k \end{matrix} \right]_q = \sum_{\lambda \subseteq k \times l} q^{|\lambda|}$$

computes the generating function for the set of Young diagrams constrained to a $k \times \ell$ box. Take the limit as $k, l \rightarrow \infty$, so that we're summing over all Young diagrams, and expand out the q -binomial coefficient. □

A better proof. We can encode our partition $\lambda = (\lambda_1 \geq \lambda_2 \geq \dots)$ with a different set of integers. Let m_i be the number of times i appears in λ (known as the **multiplicity** of i) – that is, the number of j s such that $\lambda_j = i$. Then

we can encode the **multiplicities** as $\lambda = (1^{m_1} 2^{m_2} \dots)$. For example, the partition $(6, 4, 4, 3, 1, 1)$ is now encoded as $(1^2 2^0 3^1 4^2 5^0 6^1)$, which corresponds to picking out the q^2 term in $(1 + q + q^2 + \dots)$, the 1 term in $(1 + q^2 + q^4 + \dots)$, the q^3 term in $(1 + q^3 + q^6 + \dots)$, and so on. In general, we can write the sum as

$$\sum_{n \geq 0} p(n)q^n = \sum_{m_1, m_2, \dots \geq 0} q^{m_1 + 2m_2 + 3m_3 + \dots},$$

which can then be factored as

$$\sum_{m_1} q^{m_1} \cdot \sum_{m_2} q^{2m_2} \cdot \sum_{m_3} q^{3m_3} \dots = \frac{1}{1-q} \cdot \frac{1}{1-q^2} \dots,$$

as desired. □

There are also some special classes of partitions that we can analyze using generating functions:

Definition 176

Let $p^{\text{odd}}(n)$ be the number of partitions of n into odd parts (meaning that our partition $\lambda = (\lambda_1 \geq \lambda_2 \geq \dots)$, where all λ_i are odd).

Switching to the point of view of multiplicities, we are constrained to $m_i = 0$ for all even i , so the generating function is

$$\sum_{n \geq 0} p^{\text{odd}}(n)q^n = \frac{1}{1-q} \cdot \frac{1}{1-q^3} \cdot \frac{1}{1-q^5} \dots$$

Definition 177

Let $p^{\text{dist}}(n)$ be the number of partitions of n into distinct parts (meaning that our partition satisfies $\lambda = (\lambda_1 > \lambda_2 > \dots)$).

Again thinking about multiplicities, we now constrain ourselves to have $m_i \leq 1$ for all i , so the generating function is

$$\sum_{n \geq 0} p^{\text{dist}}(n)q^n = (1+q)(1+q^2)(1+q^3)\dots$$

Theorem 178 (Euler (1748))

For any n , we have $p^{\text{odd}}(n) = p^{\text{dist}}(n)$. In other words, the number of ways to partition n into odd parts is the same as the number of ways to partition n into distinct parts.

For example, we can write 5 as a sum of odd parts as $3 + 1 + 1 = 1 + 1 + 1 + 1 + 1$, and we can write it as a sum of distinct parts as $4 + 1 = 3 + 2$.

Proof. It suffices to check that the generating functions above are equal. We start with the generating function for p^{dist} and do some algebraic manipulation:

$$\begin{aligned} (1+q)(1+q^2)(1+q^3)\dots &= \frac{(1+q)(1-q)}{1-q} \cdot \frac{(1+q^2)(1-q^2)}{1-q^2} \cdot \frac{(1+q^3)(1-q^3)}{1-q^3} \dots \\ &= \frac{(1-q^2)(1-q^4)(1-q^6)\dots}{(1-q)(1-q^2)(1-q^3)\dots} = \frac{1}{(1-q)(1-q^3)(1-q^5)\dots}, \end{aligned}$$

which is the generating function for p^{odd} as desired. □

There is also a combinatorial proof (constructing a bijection between the two sets), but that's left as an exercise to us. We'll finish today with another look at the generating function for $p(n)$ from the perspective of odd and even parts:

Theorem 179 (Euler's pentagonal number theorem (conjectured 1741, proved 1750))

We can write

$$\frac{1}{\sum_{n \geq 0} p(n)q^n} = (1 - q)(1 - q^2)(1 - q^3) \cdots = \sum_{m=-\infty}^{\infty} (-1)^m q^{m(3m-1)/2}.$$

We'll discuss this more next time (after spring break), but the q^n -coefficient of this equation basically counts the number of distinct-part partitions of n with an even and odd number of parts and finds their difference; it turns out that this is 0 for almost all values of n . The numbers of the form $m(3m - 1)/2$ are called **pentagonal numbers**, because they're the number of dots in successive dilations of a pentagon!

21 April 1, 2019

We started talking about **partition theory** last time: letting $p(n)$ be the number of partitions of n , we can write down the generating function $\sum_{n \geq 0} p(n)q^n = \frac{1}{(1-q)(1-q^2)(1-q^3)\cdots}$. We stated Theorem 179 last time, which analyzes the reciprocal of that generating function and claims that it is equal to

$$f(q) = \sum_{m=-\infty}^{\infty} (-1)^m q^{m(3m-1)/2} = 1 - q - q^2 + q^5 + q^7 - q^{12} - q^{15} + q^{22} + q^{26} + \cdots.$$

This result may be surprising, because a priori we may expect that there are many different possible coefficients when we expand out the infinite product. Instead, it turns out that most terms end up having complete cancellation, and a few others have ± 1 s! (Note that the first few pentagonal numbers for positive m are 1, 5, 12, 22; half of the terms in the infinite polynomial above come from negative m .)

Some other powers of the generating function of $p(n)$ also look nice:

Theorem 180 (Gauss)

We have

$$(f(q))^3 = \sum_{m=-\infty}^{\infty} (-1)^m \cdot m \cdot q^{m(m+1)/2}.$$

Remark 181. Unfortunately, trying to expand out $f(q)^2$ is a total mess. There isn't such a simple expression for $f(q)^2$, and there's a deep representation theoretic reason for that.

When setting up these generating functions last lecture, we also discussed different ways to represent a partition: we can either write $\lambda = (\lambda_1, \lambda_2, \dots)$ with parts in non-increasing order, or we can write $\lambda = (1^{m_1} 2^{m_2} \dots)$ in terms of its multiplicities, where m_i is the number of parts of λ equal to i . Thus, if we expand out $f(q)$, we find that

$$f(q) = (1 - q)(1 - q^2) \cdots = \sum_{m_1, m_2, \dots \in \{0,1\}} (-1)^{\sum m_i} q^{m_1 + 2m_2 + 3m_3 + \dots},$$

where $m_i \in \{0, 1\}$ means that each i can only be included at most once in the partition, $m_i = 0$ (not including i) corresponds to picking 1 in the product, and $m_i = 1$ (including i) corresponds to picking q^i . This allows us to rewrite

as

$$= \sum_{\substack{\text{partitions } \lambda \\ \text{with distinct parts}}} (-1)^{\text{parts in } \lambda} q^{|\lambda|}.$$

This lets us write Euler's pentagonal number theorem slightly differently:

Theorem 182 (Euler's pentagonal number theorem, restated)

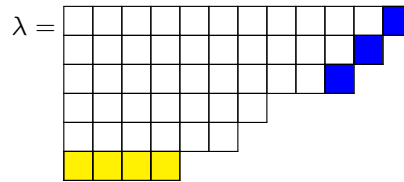
Let $p_{\text{dist}}^{\text{even}}(n)$ (resp. $p_{\text{dist}}^{\text{odd}}(n)$) be the number of partitions of n with distinct parts and an even (resp. odd) number of parts. Then we have the identity

$$p_{\text{dist}}^{\text{even}}(n) - p_{\text{dist}}^{\text{odd}}(n) = \begin{cases} (-1)^n & n = \frac{m(3m-1)}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

To prove this, we wish to set up a matching between partitions with an even and odd number of parts, and (almost always) perfectly pair them.

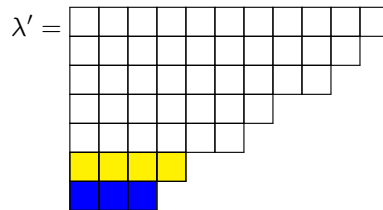
Proof by Franklin, 1881. We'll construct an **involution**, which is a function f that squares to the identity. (In other words, f is its own inverse.) Specifically, we will construct an involution σ on **almost all** partitions of n with distinct parts, such that σ sends a partition with an odd number of parts to an even number of parts and vice versa. Rigorously, if $\mu = \sigma(\lambda)$, then the number of parts in μ and λ should have different parity.

To do this, we'll define σ so that it always either adds or removes a part to λ . For example, consider the partition λ below for illustration:



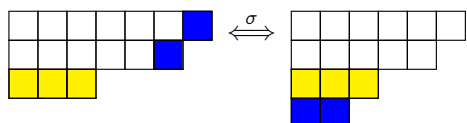
Label the yellow part (the last row) as A , and label the blue part (the longest diagonal segment starting from the top right corner) as B , and define $a = |A|$, $b = |B|$ (in this case, $a = 4$, $b = 3$). Remember that the diagonal segment from the top right corner is the rightmost part of the partition, because all parts of λ are distinct. Here's how we'll construct σ :

- If $a > b$, like in the example above, we remove the diagonal segment B and add its boxes as a new row with b boxes:

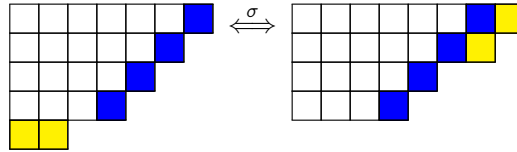


- If $a \leq b$, remove the last row and add a new diagonal segment to the right of B . For example, the map from λ' back to λ will work.

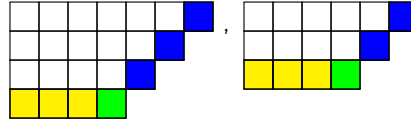
For illustration, we'll show a few more examples. For $\lambda = (7, 6, 3)$, the involution maps the following two Young diagrams to each other:



Meanwhile, for $\lambda = (7, 6, 5, 4, 2)$, we map the following two Young diagrams to each other:



However, there are a few special cases where our involution does not work (and this accounts for the ± 1 terms that we have in the generating function). Specifically, we have a problem if the yellow and blue segments, A and B , overlap with each other. It turns out that we can actually still construct an involution even in these cases, **except** when $a = b$ or $a = b + 1$. The cases $(a, b) = (4, 4)$ and $(4, 3)$ are shown below:



We can check that moving either A or B does not give us a valid partition with distinct parts, and we can call these the “pentagonal cases.” Counting the number of squares indeed gives us the pentagonal numbers $\frac{m(3m-1)}{2}$, as desired. \square

We can actually generalize Euler’s pentagonal number formula and Gauss’ formula into a single identity:

Theorem 183 (Jacobi’s triple product identity (1829))

We have the equation

$$\prod_{n \geq 1} (1 - q^{2n})(1 + q^{2n-1}z)(1 + q^{2n-1}z^{-1}) = \sum_{r=-\infty}^{\infty} q^{r^2} z^r.$$

Corollary 184

Theorem 183 has many special cases:

- If we set $z = -x^{1/2}$ and $q = x^{3/2}$, then the left hand side becomes the left side of Euler’s theorem, and we recover Euler’s pentagonal number formula.
- If we set $z = -x$, $q = x^{1/2}$, we recover Gauss’ formula.
- If we set $z = -1$, we find that

$$\prod_{m \geq 1} \frac{1 - q^m}{1 + q^m} = \sum_{r=-\infty}^{\infty} (-1)^r q^{r^2}.$$

Proof sketch. Substitute $q \rightarrow q^{1/2}$ and $z \rightarrow qz$. Then the triple product identity can be equivalently written as (moving the $\prod(1 - q^{2n})$ to the other side)

$$\prod_{n \geq 1} (1 + zq^n) \prod_{n \geq 1} (1 + z^{-1}q^{n-1}) = \left(\sum_{r=-\infty}^{\infty} z^r q^{r(r+1)/2} \right) \cdot \frac{1}{\prod_{n \geq 1} (1 - q^n)}.$$

We will interpret this identity combinatorially. The first term on the left side counts partitions with distinct parts, with z keeping track of the number of parts. So the coefficient of z^a for the first term on the left side is the generating function $\sum_{\mu} \text{partition with } a \text{ distinct parts } q^{|\mu|}$, and similarly, the coefficient of z^{-b} for the second part is $\sum_{\nu} \text{partition with } b \text{ distinct parts } q^{|\nu|-b}$.

$-b$ coming from the fact that our product has q^{n-1} instead of q^n . Next time, we'll see how we can relate these two partitions λ, ν to a more general partition λ . □

22 April 3, 2019

Recall that last lecture, we formulated the Jacobi triple product identity, which can be written equivalently in the following form:

$$\prod_{n \geq 1} (1 + zq^n) \prod_{n \geq 1} (1 + z^{-1}q^{n-1}) = \sum_{r=-\infty}^{\infty} z^r q^{r(r+1)/2} \prod_{n \geq 1} \frac{1}{1 - q^n}.$$

We'll continue the combinatorial proof of this identity today, coming up with a combinatorial interpretation of each term in terms of certain partitions.

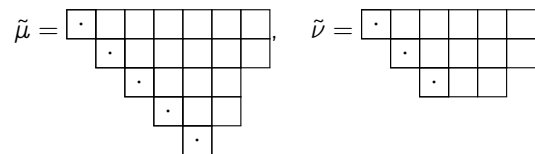
Proof, continued. We analyze the z -coefficients. On the left-hand side, the first product's coefficient of z^a is $\sum_{\substack{\mu \text{ partition} \\ \text{with } a \text{ distinct parts}}} q^{|\mu|}$, and the second product's coefficient of z^{-b} is $\sum_{\substack{\nu \text{ partition} \\ \text{with } b \text{ distinct parts}}} q^{|\nu|-b}$. Meanwhile, the product on the right is the usual generating function for all partitions, so we want to somehow we want to combine partitions together – that is, we want to find a bijection

$$(a, b, \mu, \nu) \rightarrow (r, \lambda),$$

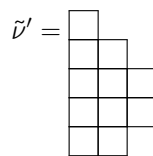
where μ has a distinct parts and ν has b distinct parts. We also need to make sure the monomials (powers of z and q) match up when we construct this bijection: looking at powers of z , we need $a - b = r$, and looking at powers of

$$q, \text{ we need } |\mu| + |\nu| - b = \frac{r(r+1)}{2} + |\lambda|.$$

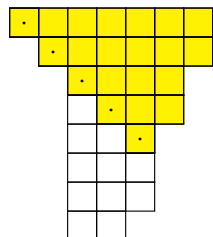
For this bijection, since μ and ν have distinct parts, we can represent them with **shifted Young diagrams** instead of standard ones. For example, if $\mu = (7, 6, 4, 3, 1)$ and $\nu = (6, 5, 3)$, then our Young diagrams look like



The construction is to transpose $\tilde{\nu}$ and remove its $b = 3$ diagonal boxes:



Now, take $\tilde{\mu}$ and $\tilde{\nu}'$ and glue them together, so that the removed boxes of $\tilde{\nu}'$ would have overlapped with the last dots of $\tilde{\mu}$:



With this, we can now chop off the first $a - b = 2$ columns off, so that we have a standard Young diagram again.

chopped-off triangle has $1 + 2 + \dots + (b - a - 1)$ boxes. Thus the glued shape has $|\lambda| + \frac{(-r-1)(-r)}{2} = |\lambda| + \frac{r(r+1)}{2}$ squares, and the identity indeed still holds.

Finally, we explain why this construction $(a, b, \mu, \nu) \rightarrow (r, \lambda)$ is a bijection. If we're given r and λ : attach a small triangle to either the left or top of λ (depending on the sign of r), and then chop the shape along the diagonal. This gives us $\tilde{\mu}$ and $\tilde{\nu}'$, from which we can recover μ and ν (add back the diagonal by putting as many boxes as possible), which tells us our original partitions and a and b as well. This finishes the proof and gives a combinatorial explanation for the Jacobi triple product identity. \square

With that, we'll move on to the next topic of this class. We've been talking about Young diagrams and partitions a lot so far, so in the second half of 18.212, we'll start to discuss **graphs, networks, and trees**.

Definition 185

Graphs are objects with **vertices** (also **nodes**) and **edges** that connect pairs of vertices. A **labeled tree** is a connected graph on n nodes with no cycles, where the nodes are labeled $1, 2, \dots, n$. An **unlabeled tree** is the analogous object but where nodes are not labeled.

Example 186

There are 3 labeled trees on 3 vertices – place the numbers 1, 2, 3 at the vertices of an equilateral triangle, and remove one of the edges. However, all of these labeled trees correspond to the same unlabeled tree (a path on three vertices).

We actually have an explicit formula for counting trees:

Theorem 187 (Cayley's formula)

The number of labeled trees on n nodes is n^{n-2} .

This quantity n^{n-2} actually appears in mathematics fairly often – they are probably the next most famous set of numbers after Catalan numbers, and there are many combinatorial interpretations attached to them.

Remark 188. *In mathematics, there's a "law" that mathematical formulas are never named after the people who actually discover them. Cayley wrote a paper on this result in 1889, but the same result was previously given by Borchard in 1860 and Sylvester in 1857.*

Cayley didn't actually give a complete proof in his paper. Instead of presenting the older proofs, Professor Postnikov will show his favorite proof, which is probably the shortest one.

Proof by Rényi, 1967. We're going to use induction (!), proving a more general result (because it's hard to get n^{n-2} from other terms of the form k^{k-2}):

Definition 189

For any tree T , define the **weight** of T to be the monomial $x^T = x_1^{\deg(1)-1} x_2^{\deg(2)-1} \dots x_n^{\deg(n)-1}$ (where the **degree** $\deg(i)$ of i is the number of edges that have i as an endpoint).

The claim that we will prove is the following:

Claim 190. *We have the identity*

$$F_n(x_1, \dots, x_n) = \sum_{\substack{\text{labeled trees } T \\ \text{on } n \text{ vertices}}} x^T = (x_1 + \dots + x_n)^{n-2}.$$

Cayley's formula directly follows by setting all $x_i = 1$. The idea is that this inductive hypothesis is stronger than what we had before, so the result will be easier to show.

To show the identity, we induct on n . Define $R_n = F_n - (x_1 + \dots + x_n)^{n-2}$; our goal is to show that $R_n = 0$. The base cases $n = 1, 2$ are easy, and for the inductive step, we make the following observations:

- **R_n is a polynomial in x_1, \dots, x_n of degree at most $n - 2$.** This is because the second term has degree $n - 2$, and every monomial x^T has degree $n - 2$. Indeed, there are $n - 1$ edges in any tree of n vertices, each of which adds 2 to the degree of the weight (one for each endpoint), but the -1 s in the definition of x^T bring down the degree by n , for a total degree of $2(n - 1) - n = n - 2$.
- **For any $1 \leq i \leq n$, if we set $x_i = 0$, R_n evaluates to 0.** To prove this, we can assume $i = n$ by symmetry. Because the degree of x_n is $\deg(n) - 1$ in any tree T , when we set $x_n = 0$, the only terms that remain are from trees where n is a leaf (a vertex of degree 1). In other words, F_n evaluated at $x_n = 0$ is

$$F_n(x_1, \dots, x_{n-1}, 0) = \sum_{T: n \text{ leaf}} x^T = \sum_{T' \text{ with } n-1 \text{ nodes}} x^{T'} \cdot (x_1 + x_2 + \dots + x_{n-1})$$

since for any tree T' , we can connect n to any of the other vertices, which keeps the exponent of x_n at 0 but adds 1 to the exponent of some x_i . But by the induction hypothesis, this sum is $F_{n-1}(x_1 + \dots + x_{n-1})$, so

$$\begin{aligned} R_n|_{x_n=0} &= F_{n-1}(x_1 + \dots + x_{n-1})(x_1 + x_2 + \dots + x_{n-1}) - (x_1 + x_2 + \dots + x_{n-1} + 0)^{n-2} \\ &= (x_1 + x_2 + \dots + x_{n-1})^{n-2} - (x_1 + x_2 + \dots + x_{n-1})^{n-2} \\ &= 0. \end{aligned}$$

- But now we're done: since $R_n(x_1, \dots, x_n)$ is a polynomial on n variables of degree less than n , and x_1, x_2, \dots, x_n are all factors, then the polynomial must be 0.

□

This may almost feel like cheating, because we've used essentially no properties of trees at all in the proof. We'll talk a bit more about this next time!

23 April 5, 2019

Recall that we started talking about the Cayley formula last week (which states that the number of labeled trees on n vertices is n^{n-2}). In last lecture's proof, we considered the polynomial

$$F_n(x_1, \dots, x_n) = \sum_{T \text{ tree on } [n]} \prod_{i=1}^n x_i^{\deg(i)-1},$$

and we showed that $R_n = F_n - (x_1 + \dots + x_n)^{n-2}$ is identically zero. Let's review the argument again: we induct on n (the base case $n = 1, 2$ is easy). First, R_n is either a homogeneous polynomial in our variables x_1, \dots, x_n of degree $n - 2$ or zero (because each monomial in the sum has total degree $\sum(\deg(i)) - n = 2(n - 1) - n = n - 2$). Then, if

we pick any $1 \leq i \leq n$ and evaluate R_n at $x_i = 0$, then the result is

$$R_{n-1}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, n)(x_1 + \dots + x_{i-1} + x_{i+1} + \dots + x_n),$$

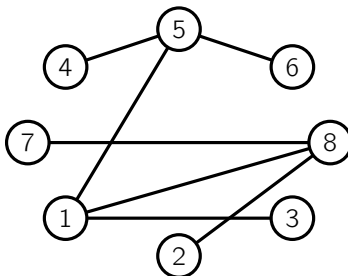
because all terms disappear except those where i is a leaf (a vertex of degree 1), and then in those cases we can attach i to any vertex of a labeled tree on the remaining vertices. The right hand side is 0 by induction, so that means that f when evaluated at (any) $x_i = 0$ is always 0. So x_i is a factor for all i , meaning R_n is a multiple of $x_1 x_2 \dots x_n$, which can only occur if $R_n = 0$. Setting all $x_i = 1$, we indeed find that the number of trees on n vertices is n^{n-2} .

We barely used any properties of trees here, so this proof alone may be difficult to parse! In combinatorics, we like bijective proofs, so it would be nice to construct a bijection between labeled trees and some other set of n^{n-2} elements. We'll go through two of them in today's class.

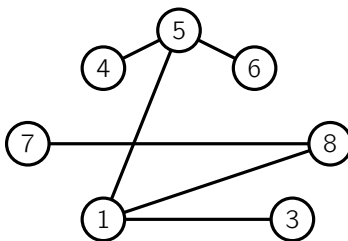
Proof by Prufer (1918). We'll set up a bijection between trees T on n vertices and sequences (c_1, \dots, c_{n-2}) , where all $c_i \in [n]$, known as the **Prufer code**. (Clearly, the latter set has n^{n-2} total elements, so this would show the desired result.) We'll write out the procedure and then do an example.

1. Any tree with at least 1 vertex contains a leaf. Find the vertex v with the minimal label,
2. Record (in our code) the vertex in our tree that v is attached to (by its adjacent edge),
3. Remove v (and the adjacent edge) from T .
4. Repeat steps 1–3 a total of $n - 2$ times.

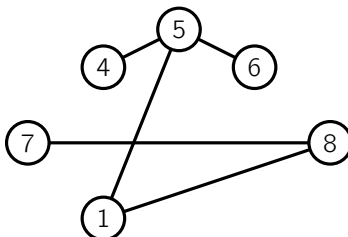
As an example, consider the tree below:



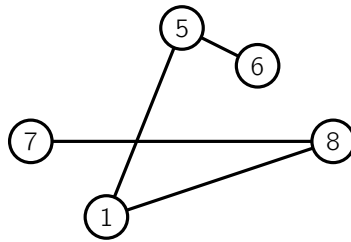
Initially, the smallest leaf has label 2, and it is attached to 8, so we write down 8 and erase label 2.



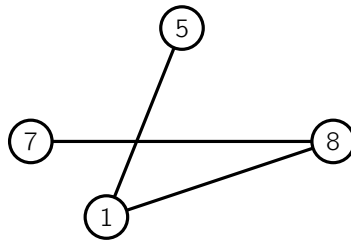
Now, the minimal leaf is 3, and it is attached to 1, so we write down 1.



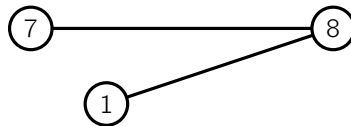
After that, the minimal leaf is 4, and it is attached to $\boxed{5}$:



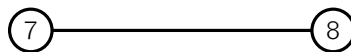
6 is attached to $\boxed{5}$:



5 is attached to $\boxed{1}$:



Finally, 1 is attached to $\boxed{8}$:

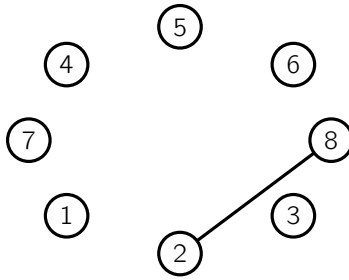


That completes our algorithm, and our final Prufer code is $\boxed{(8, 1, 5, 5, 1, 8)}$. To show this is actually a bijection, we need to be able to invert our construction. Notice that every vertex v of the tree appears $\deg(v) - 1$ times in the final Prufer code (for example, 5 appears twice in the code and started with degree 3). After all, we record i in the Prufer code every time an edge to it is erased, except for the last edge (at which point it is removed and its neighbor is recorded instead, or the algorithm has terminated).

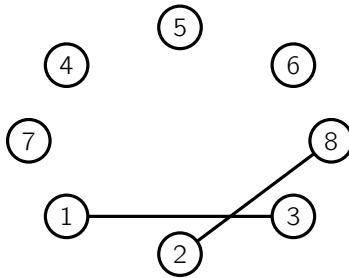
In particular, the labels that don't show up in the Prufer code are exactly the original leaves in our tree! With that, we can now describe the decoding process:

1. Initially, we are given a **Code** (c_1, \dots, c_{n-2}) . Initialize a set of **Labels** as $\{1, 2, \dots, n\}$.
2. Find the minimal leaf by finding the smallest element ℓ of **Labels** that is not in **Code**. Connect a node containing ℓ to the first element c of **Code** (this was the first step of our coding process, removing the smallest leaf and recording down its neighbor).
3. Remove ℓ from **Labels** and c from **Code**.
4. Repeat steps 1–3 a total of $n - 2$ times until **Code** is empty.
5. Finally, there should be two elements left in **Labels**, and we connect them.

Let's verify that this works by example: start with the Prufer **Code** $(8, 1, 5, 5, 1, 8)$ and **Labels** $\{1, 2, 3, 4, 5, 6, 7, 8\}$. The minimal element that doesn't appear is 2, so first we connect 2 to 8.



After removing 8 from **Code** and 2 from **Labels**, **Code** is (1, 5, 5, 1, 8) and **Labels** is {1, 3, 4, 5, 6, 7, 8}. Then the smallest element of **Labels** not in **Code** is 3, so we should connect 3 to 1:



Continuing this process, we indeed add the edges that were removed during the forward algorithm one by one, in the order that they were removed: the next steps are to connect 4 to 5, then 6 to 5, then 5 to 1, and 1 to 8. Finally, **Labels** still contains {7, 8}, so we connect those two nodes to finish the tree. □

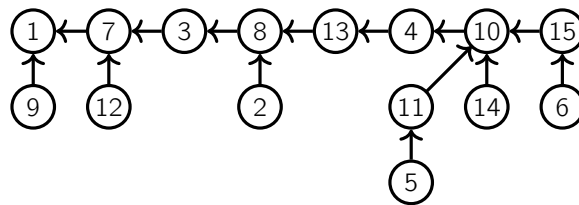
The Prufer code encoding tells us some additional information about the structure of our trees:

Proposition 191

Fix $d_1, \dots, d_n \geq 1$. Then the number of labeled trees on n vertices with degrees d_1, \dots, d_n for vertices $1, 2, \dots, n$ (where $d_1 + \dots + d_n = 2n - 2$) is the multinomial coefficient $\binom{n-2}{d_1-1, d_2-1, \dots, d_n-1}$.

Finally, we'll show a more recent bijective proof:

Proof by Eğecioğlu-Remmel (1986) based on Joyal (1981). Start with a labeled tree, and “root the tree at 1” – in other words, orient all edges of our graph towards the root. For illustration, we'll consider the tree below:



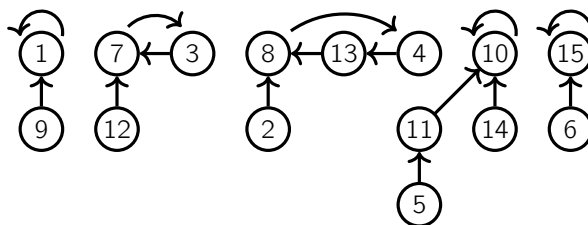
Consider the path from the maximal element $n = 15$ to 1, and note down the elements along the way:

$$P = (15, 10, 4, 13, 8, 3, 7, 1).$$

We mark the set of left-to-right minima (that is, the elements that are the smallest so far in P – in this case, it's 15, 10, 4, 3, and 1). We then rewrite P as a permutation on the path elements by starting a new cycle at each left-to-right minimum:

$$P' = (15)(10)(4, 13, 8)(3, 7)(1).$$

The bijective step is now to **replace the path from n to 1 with the collection of cycles P'** . In particular, our example tree from above now looks as shown below:



Notice that the new graph will always have a loop at 1 and a loop at 15. After we do this process, there will be exactly one edge coming out of each vertex (because this was true for all vertices except 1 to start, and then rewriting the path as a combination of cycles creates a permutation on those path vertices), so we can define a function f by setting $f(i) = j$ if i points to j . This is a map from $[n]$ to $[n]$ that fixes 1 and n , so there are exactly n^{n-2} such functions. To recover the original tree, we find the set of cycles in our function, cyclically orient them so that the smallest element comes first, and then concatenate them from largest first element to smallest – that gives us the original path in our tree, and we recover the tree by replacing the set of cycles with our path. \square

24 April 8, 2019

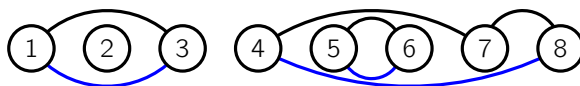
Our second problem sets were returned to us today, so we'll be doing some more student presentations.

Problem 192

Show that the number of noncrossing partitions of $\{1, 2, \dots, n\}$ is the Catalan number C_n .

We'll show two solutions for this problem:

Solution by Vanshika Jain. We construct a bijection between a Dyck path of length $2n$ and a noncrossing partition of $[n]$. As an example, consider the partition $(13|2|478|56)$ for $n = 8$. The idea is that we want every element to correspond to 2 letters in our Dyck path, so for each part of the partition with more than 1 element, draw an extra line from the first to last number in the partition (so now we have closed cycles):

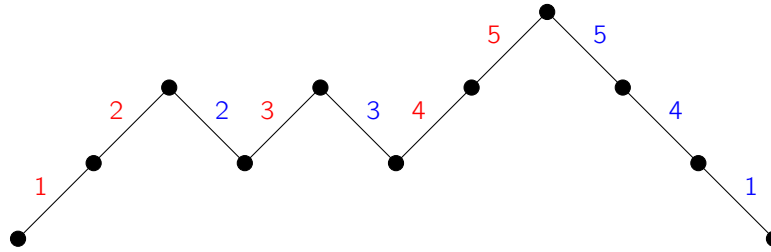


Now, we construct our Dyck path by looking at the elements $i \in \{1, 2, \dots, n\}$ in order:

- If there are two arcs from i that go to the right, we write **UU**,
- if there are two arcs from i to the left, we write **DD**,
- if there are no arcs from i , we write **UD**, and
- if there is one arc in each direction from i , we write **DU**.

For example, in the above diagram, we get the Dyck path **UUUDDDUUUUDDDUDD**. This is a Dyck path on **Us** and **Ds**, because **Ds** correspond to ending arcs and **Us** to starting them (if we think of “no arc” as a self-loop). This is a bijection because we can break up any Dyck path of length $2n$ into n two-character sequences, and then we draw arcs by taking **Ds** from left to right and connecting them to the nearest **U** before them. Then deleting the long edges from cycles and removing self-loops gives us the original set-partition. \square

Solution by Sanzeed Anwar. We'll construct a bijection in the opposite direction. Start with a Dyck path of length $2n$, and mark all of the "up" steps in order from 1 to n . Then we can label the "down" steps by matching steps at the same level – for example, here is the labeling corresponding to the Dyck path **UUDUDUUDDD**:



Our partition then consists of the **maximal continual sequences of down steps** – in this case, we have the partition $(145|2|3)$. This partition will always be noncrossing, because if we have two indices i, j that are adjacent in a part of the partition, then all up steps between i and j must also be matched before j to avoid "blocking" i . So there is no way to connect an element $k \in (i, j)$ to an element $\ell > j$, meaning no crossing $i < j < k < \ell$ exists.

To reverse this map, arrange the parts of a partition in descending order of largest element, and so that the elements decrease within each part. Then insert the part of the partition $\{a_1 > a_2 > \dots\}$ immediately after the a_1 th up step. This always yields a valid Dyck path because only down steps labeled k or less can show up before the k th up step. □

Problem 193

Find a closed formula for the number of saturated chains from the minimal element $\hat{0} = (1|2|\dots|n)$ to the maximal element $\hat{1} = (1 \dots n)$ in the partition lattice Π_n .

Solution by Congyue Deng. Following the path from $\hat{0}$ to $\hat{1}$ along the saturated chain corresponds to repeated merging of two parts into one: for example, we can go from $(1|2|3|4|5)$ to $(12|3|4|5)$ to $(124|3|5)$ to $(124|35)$ to (12345) . This process always takes $(n - 1)$ steps, because we decrease the number of parts from n to 1. Furthermore, on the k th step, there are $\binom{n-k+1}{2}$ ways to merge blocks, since we have $n - k + 1$ blocks. Thus, our answer is

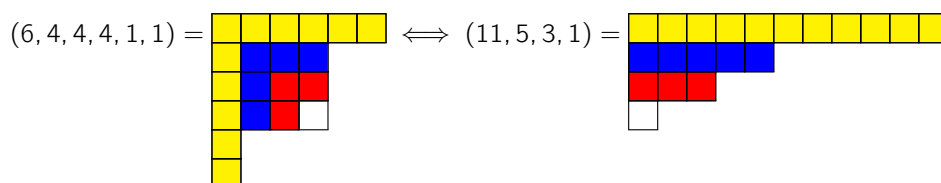
$$\prod_{k=1}^{n-1} \binom{n-k+1}{2} = \frac{(n-1)!n!}{2^{n-1}}$$

(by expanding out the binomial coefficients and multiplying each term across all k). □

Problem 194

Find a bijection between partitions with odd distinct parts and self-conjugate partitions.

Solution by Sophia Xia. Consider the Young diagram representation of a partition, and bend each odd/distinct partition at the middle and join them together along the diagonal, as shown below by example:



Since partitions with odd distinct parts differ by at least 2, the lengths of the left “arms” of each color will differ by at least 1, meaning that later arms will not extend farther than earlier ones and that we do have a valid Young diagram. This procedure is reversible, so we do have a bijection, showing the desired result. \square

Problem 195

Find the number of paths in Young’s lattice of length $2n$ from $\hat{0}$ back to itself (in any sequence of up and down steps).

Solution by Yogeshwar Velingker. We wish to consider the set of words of length $2n$ with n **U**s and n **D**s and compute the coefficient of $\hat{0}$ in $\sum_{W \text{ word}} W\hat{0}$. Recall that for a sequence of **U**s and **D**s, such as $DDUDUU\hat{0}$, each **U** needs to be matched with a **D** on its left: But if we have an arbitrary word, that just means we can match any pair of indices that we want; in addition, any matching of n pairs corresponds to a specific word, because the smaller elements of the pairs are the **D**s and the larger elements are the **U**s. So this problem reduces to finding the number of ways to pair $\{1, 2, \dots, 2n\}$ into n pairs: there’s $(2n - 1)$ ways to match 1, then $(2n - 3)$ ways to match the next number in the list that hasn’t been matched, then $(2n - 5)$, and so on, which yields a final answer of $(2n - 1)!!$. \square

Problem 196

Show that the Bell number (that is, the number of partitions of n into sets) is given by

$$B(n) = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}.$$

Solution by Wanlin Li. Recall that $B(n) = \sum_{m=0}^n S(n, m)$, where $S(n, m)$ is the number of ways to partition n into m subsets. From class, we know that

$$x^n = \sum_{m=0}^n S(n, m)(x)_m,$$

where $(x)_m$ denotes the falling power of x . Replacing x by k and taking constraints on S and the binomial coefficient into account, we find that

$$k^n = \sum_{m=0}^n S(n, m) \binom{k}{m} m! = \sum_{m=0}^k S(n, m) \binom{k}{m} m!,$$

Plugging this into the right-hand side, we have

$$\frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!} = \frac{1}{e} \sum_{k=0}^{\infty} \sum_{m=0}^k \frac{S(n, m) \binom{k}{m} m!}{k!}.$$

Expanding the binomial coefficient and switching the order of summation yields

$$= \frac{1}{e} \sum_{k=0}^{\infty} \sum_{m=0}^k \frac{S(n, m)}{(k - m)!} = \frac{1}{e} \sum_{m=0}^{\infty} \sum_{k=m}^{\infty} \frac{S(n, m)}{(k - m)!} = \sum_{m=0}^{\infty} S(n, m) = \sum_{m=0}^n S(n, m),$$

which is exactly the Bell number $B(n)$, as desired. \square

Problem 197

Find a bijection between distinct partitions of n and odd partitions of n .

Solution by Sarah Wang. Suppose we start with a partition $n = \lambda_1 + \cdots + \lambda_n$ with all distinct parts, and factor out the powers of 2 from each part, writing $\lambda_i = \lambda_i^{\text{odd}} 2^{m_i}$. Then the corresponding partition of n with odd parts includes each λ_i^{odd} with multiplicity 2^{m_i} (preserving the sum, since the 2^{m_i} copies of λ_i^{odd} add up to λ_i). For example, $13 = 5+4+3+1$ becomes $13 = 5 + 3 + 1 + 1 + 1 + 1 + 1$.

For the reverse procedure, start with an odd partition $n = \lambda_1^{a_1} \lambda_2^{a_2} \cdots$, where a_i is the multiplicity of each λ_i . Now expand each a_i (uniquely, using the base-2 representation) as a sum of powers of 2, and replace $\lambda_1^{a_1}$ with $\sum \lambda_1 \cdot 2^m$, where $\sum 2^m = a_1$. For example, $22 = 5 + 5 + 5 + 3 + 3 + 1$ becomes $22 = 10 + 6 + 5 + 1$. \square

Problem 198

Construct a non-recursive description of the Fibonacci lattice.

Solution by Chiu Yu Cheng. First of all, note that the number of compositions of n (partitions of n , but where order matters) with all parts equal to 1 or 2 is the Fibonacci number F_{n+1} , because we either start a composition with 1 or 2 and then fill in the rest in one of F_n or F_{n-1} ways (respectively) by induction. (The base cases $n = 1, 2$ are clear.) Consider a graph where every vertex is a **composition of some nonnegative integer** n with only 1s and 2s, where two vertices v_1, v_2 are connected (in poset language, $v_1 \leq v_2$) if v_2 is obtained by v_1 by either (1) inserting a 1 anywhere to the left of the leftmost 1 in v_1 , or (2) changing the leftmost 1 to a 2. In other words, v_1 is obtained from v_2 by either (1*) removing the leftmost 1 or (2*) changing a 2 to a 1, when 2 is on the left of the leftmost 1.

We claim this graph is isomorphic to the Fibonacci lattice. Two vertices are only connected if the sum of the composition parts differ by 1, so we can define the rank to be that sum. From here, we just need to check that the recursive definition of the Fibonacci lattice is satisfied (in which we flip the elements from rank $(n - 1)$ up, and then add a new edge from each element of rank n). Indeed, the elements that start with 1 of rank $(n + 1)$ are only connected to the corresponding elements of rank n with that 1 removed (because we couldn't have used operation (2) to get to them), and the elements that start with a 2 correspond to the elements of rank $(n - 1)$ with that initial 2 removed. This latter claim is a bit harder to verify: if we have a composition w of $(n - 1)$ and the corresponding composition $2 + w$ of $(n + 1)$, notice that w is connected to a composition w' of n through operation (1) (resp. (2)) if and only if applying operation (2) (resp. (1)) to n can give us $2 + w$.

Finally, we define the meet and join operators more explicitly. Note that $a \leq b$ if and only if there is a sequence of (1) and (2) steps from a to b . We define the "join" operation as follows: if either of x and y start with 1, then it has a unique lower neighbor, meaning that $(1 + x) \vee y = x \vee y$ (and similarly $x \vee (1 + y) = x \vee y$). Meanwhile, if both x and y start with a 2, the subgraph with all compositions starting with 2 is isomorphic to the initial graph, so $(2 + x) \vee (2 + y) = x \vee y$. The join operation is therefore defined with these recursive relations and $x \vee x = x$. On the other hand, for "meet," notice that every composition of n terms sits below $2n = 2 + \cdots + 2$, so a composition with n terms and a composition with m terms have a common upper neighbor $2 \max(m, n) = 2 + 2 + \cdots + 2$. We can then look at the finitely many elements of the Fibonacci lattice below it and find the one that is minimal to obtain $x \wedge y$. \square

25 April 10, 2019

We'll start today with a few more problem presentations before moving on to the next topic.

Problem 199

Show that the Bell numbers $B(0) = 1, B(1) = 1, B(2) = 2, B(3) = 5, B(4) = 15, B(5) = 52, \dots$ can be calculated as the first (and also last) entries in rows of the following triangle:

$$\begin{array}{ccccccc} & & & & & & 1 \\ & & & & & & 1 & 2 \\ & & & & & 2 & 3 & 5 \\ & & & 5 & 7 & 10 & 15 \\ & 15 & 20 & 27 & 37 & 52 \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

In this triangle, the first number in each row (except the first row) equals the last number in the previous row, and any other number equals the sum of the two numbers to the left and above it.

Solution by Fadi Atieh. We'll prove this by induction; the base case is easy because $B_0 = B_1 = 1$. For the induction step, assume that we have already built k rows of this triangle. We know that the leftmost entry of the k th row is B_{k-1} , and the rightmost entry is B_k , so the $(k+1)$ th row begins with B_k . The next entry is $B_k + B_{k-1}$, and in general we can use the Bell triangle definition to recursively push any entry up and to the left until we reach a sum of terms on the left diagonal.

we just need to figure out the actual recurrence relation. Notice that B_{k+1} , the rightmost entry in the $(k+1)$ th row, is the sum of the term above it and to its left, each of which is the sum of the terms above them and to their lefts, repeating until we get to one of the leftmost diagonal entries B_i . The total number of ways we can reach B_i is the number of ways to go up $(k-i)$ times and left i times, so we arrive at

$$B_{k+1} = \sum_{i=0}^k \binom{k}{i} B_i.$$

This is indeed the recursive formula for Bell numbers: we pick $k-i$ elements out of $\{1, \dots, k\}$ to be in the same partition as $(k+1)$ in one of $\binom{k}{k-i} = \binom{k}{i}$ ways, and then we partition the remaining numbers in one of B_i ways. Thus by induction, B_{k+1} is indeed the next Bell number, as desired. \square

Problem 200

Show that the Stirling numbers satisfy the recurrence relations

$$S(n+1, k) = kS(n, k) + S(n, k-1), \quad c(n+1, k) = nc(n, k) + c(n, k-1).$$

Solution by Ramya Durvasula. Recall that $S(n, k)$ counts the number of ways to place n numbers into a set-partition with k groups. If we want to partition $[n+1]$ into k groups, we can place $(n+1)$ into its own partition. Then the remaining integers must be partitioned into $(k-1)$ groups, which can be done in $S(n, k-1)$ ways. On the other hand, $(n+1)$ may not be on its own, in which case we first partition the numbers 1 through n into k sets. Then we put $(n+1)$ into one of the k sets in one of k ways, so this process has $kS(n, k)$ possible choices. Adding these together yields $S(n+1, k) = kS(n, k) + S(n, k-1)$, as desired.

Similarly, $c(n, k)$ counts the number of permutation of n numbers with k cycles. If we are trying to compute $c(n+1, k)$, again we think about where $(n+1)$ goes: if it goes in its own cycle, we have $(k-1)$ cycles left and thus $c(n, k-1)$ ways to finish. On the other hand, if $(n+1)$ is in a cycle, first create k cycles with the numbers 1

through n in one of $c(n, k)$ ways. Then we can add $n + 1$ to one of the n spots in existing cycles (since a cycle of length ℓ has ℓ possible insertion points), accounting for the remaining $nc(n, k)$ ways. Putting these together yields $c(n + 1, k) = nc(n, k) + c(n, k - 1)$, as desired. \square

Problem 201

Define the operators $X : f(x) \rightarrow xf(x)$ and $D : f(x) \rightarrow f'(x)$. Define $f_n(x) = (X + D)^n(1)$ (for example, $f_0(x) = 1, f_1(x) = x, f_2(x) = x^2 + 1, f_3(x) = x^3 + 3x$). Find a closed-form expression for the constant term $f_n(0)$.

Solution by Mathew Ganatra. The recursive formula for these polynomials is $f_n(x) = xf_{n-1}(x) + f'_{n-1}(x)$ – in particular, plugging in $x = 0$, $f_n(0) = f'_{n-1}(0)$. For sake of finding patterns, we write out some more values of $f_n(x)$:

$$f_4(x) = x^4 + 6x^2 + 3, f_5(x) = x^5 + 10x^3 + 15x, f_6(x) = x^5 + 15x^4 + 45x^2 + 15,$$

$$f_7(x) = x^7 + 21x^5 + 105x^3 + 105x, f_8(x) = x^8 + 28x^6 + 210x^4 + 420x^2 + 105.$$

From those expressions, we can extrapolate to the formula

$$f_n(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} x^{n-2k} \frac{(n)_{2k}}{k!2^k}.$$

We prove this formula by induction: the base case can be observed from our examples above, and for the inductive step, notice that

$$\begin{aligned} f_{n+1}(x) &= (X + D) \sum_{k=0}^{\lfloor n/2 \rfloor} x^{n-2k} \frac{(n)_{2k}}{k!2^k} \\ &= \sum_{k=0}^{\lfloor n/2 \rfloor} x^{n+1-2k} \frac{(n)_{2k}}{k!2^k} + \sum_{k=0}^{\lfloor n/2 \rfloor} (n-2k)x^{n-1-2k} \frac{(n)_{2k}}{k!2^k} \\ &= \sum_{k=0}^{\lfloor n/2 \rfloor + 1} x^{n+1-2k} \left(\frac{(n)_{2k}}{k!2^k} + (n+2-2k) \frac{(n)_{2k-2}}{(k-1)!2^{k-1}} \right). \end{aligned}$$

The bounds of the sum are correct, because if n is even, then plugging in $k = \frac{n}{2} + 1$ makes the coefficient zero, so we can indeed just sum up to $\lfloor (n+1)/2 \rfloor$. Thus we just need to verify that this coefficient is correct for generic k :

$$\begin{aligned} \frac{(n)_{2k}}{k!2^k} + (n+2-2k) \frac{(n)_{2k-2}}{(k-1)!2^{k-1}} &= \frac{(n)_{2k-2}}{k!2^k} ((n-2k+2)(n-2k+1) + (n-2k+2)(2k)) \\ &= \frac{(n)_{2k-2}}{k!2^k} \cdot (n-2k+2)(n+1) \\ &= \frac{(n+1)_{2k}}{k!2^k}, \end{aligned}$$

completing the induction and proof. To extract the answer, we're only concerned with the last term of our formula for f_n : if n is odd, the constant term is $\boxed{0}$, and if n is even, our expression simplifies to

$$f_n(0) = \frac{(n)_n}{(n/2)!2^{n/2}} = \frac{n!}{2 \cdot 4 \cdots n} = \boxed{(n-1)!}.$$

\square

It turns out there is a connection between this problem and the Problem 195 – the key relations in both cases are

that the up and down operators satisfy $DU - UD = I$, $D(\emptyset) = 0$. In addition, the coefficients $\frac{\binom{n}{2k}}{k!2^k}$ in $f_n(x)$ that we've just arrived count the number of ways to take $2n$ steps in Young's lattice and end up on the k th level.

But now it's time to move on to a new topic – we'll next discuss **spanning trees of graphs**. From here on, the notation $G = (V, E)$ means that we have a graph with a set of vertices V and edges E between them. Until otherwise specified, we will not allow loops (edges where both endpoints are the same vertex), but we will allow multiple edges between the same two vertices.

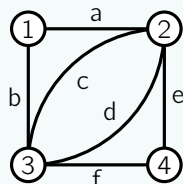
Definition 202

Let $G = (V, E)$ be a graph. A **spanning tree** of G is a connected **subgraph** $T = (V, E')$ which is also a tree. Let $T(G)$ denote the number of spanning trees of G .

In other words, to get a spanning tree T , we take a subset of the edges E such that the vertices V remain connected. Notably, T has no cycles, and repeated edges aren't allowed either (because they would form a cycle). Thus, we pick some $|V| - 1$ edges that “span” or connect all vertices of the original graph.

Example 203

Consider the graph G below:



We can check that $T(G) = 12$, because there are 4 ways to form a tree without the double edge and $2 \cdot 2 \cdot 2$ ways to form a tree that does use it.

One other case we've already studied is $T(K_n) = n^{n-2}$ (there are n^{n-2} spanning trees for the complete graph on n vertices), because any labeled tree will work. Counting spanning trees may look difficult from what we've done so far, but it turns out there is a general formula for doing this, which we'll discuss next time!

26 April 12, 2019

Today, we're going to start talking about the **matrix tree theorem**. Recall, that if we have a graph $G = (V, E)$ with no loops but potentially multiple edges, we define a **spanning tree** to be a subgraph $T = (V, E')$ (for some subset $E' \subset E$) which is a tree on the vertices V . Our goal is to find a systematic way to count these spanning trees.

For that, we'll turn to linear algebra – our first step is to associate certain matrices with graphs. For notational purposes, number the vertices V from 1 to n , and also label the edges E from 1 to $m = |E|$.

Definition 204

The **incidence matrix** of a graph $G = (V, E)$ is the $n \times m$ matrix $C = (c_{ie})$, where $c_{ie} = 1$ if i is incident to e and 0 otherwise.

For example, letting the columns correspond to the edges a, b, c, d, e, f , respectively, the incidence matrix of the

graph in Example 203 can be written as

$$C = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Definition 205

The **adjacency matrix** of a graph G is the $n \times n$ matrix $A = (a_{ij})$, where a_{ij} is the number of edges connecting i to j .

Notice that the diagonal entries of A are always 0, because we're assuming we have no self-loops. For our example, we have

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

The next definition is the one we really care about:

Definition 206

The **Laplacian matrix** of a graph G is the $n \times n$ matrix $L = D - A$, where D is the diagonal matrix with diagonal entries $d_i = \deg_G(i)$.

For our example, this gives us the Laplacian matrix

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 4 & -2 & 0 \\ -1 & -2 & 4 & 1 \\ 0 & -1 & -1 & 2 \end{bmatrix}.$$

Since we're using undirected graphs in this discussion, both the Laplacian and adjacency matrices are symmetric. Also, all row sums of L are 0 (since for a given vertex, each adjacent edge contributes +1 towards the diagonal entry in D but also contributes -1 to an off-diagonal entry in $-A$), meaning that the determinant of L is 0. We're now ready to formulate the main result:

Theorem 207 (Matrix tree theorem)

Fix $i \in \{1, \dots, n\}$, and define the **reduced Laplacian matrix**

$$\tilde{L} = L^i = L \text{ with the } i\text{th row and column removed.}$$

Then the number of spanning trees of G is equal to $\det \tilde{L}$.

In particular, this determinant does not depend on the choice of i , because of the following more general fact:

Fact 208

For any symmetric matrix $n \times n$ matrix B with row sums 0, the determinants of B^i (known as **principal cofactors** of B) are all equal. Moreover, if B 's eigenvalues are $\mu_1, \dots, \mu_{n-1}, \mu_n = 0$ (there is always a zero eigenvalue because of the zero row sum), those determinants are equal to $\frac{\mu_1 \cdots \mu_{n-1}}{n}$.

The matrix tree theorem is also called **Kirchhoff's theorem**, and it is related to Kirchhoff's laws of electricity (which were described in the 1840s). But the theorem was first proved by Borchardt in 1860, and it's also closely related to Sylvester's work from 1857. Before diving into the proof, we'll show some applications of the result:

Example 209

Continuing to follow Example 203, let's compute the number of spanning trees using the matrix tree theorem.

We'll set $i = 2$ for ease of computation (since vertex 2 has lots of edges coming out of it, and we want to keep our determinant calculation easy); the second principal cofactor (removing the second row and column) of the Laplacian matrix is

$$\tilde{L} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 2 \end{bmatrix} \implies \det \tilde{L} = 16 - 2 - 2 = 12,$$

which is the answer that we've previously derived! And we can check that other principal cofactors of L also gives us a determinant of 12.

Remark 210. *The eigenvalues of the adjacency matrix $A(G)$ form the **spectrum** of the graph G , and **spectral graph theory** studies properties of graphs in terms of this spectrum. However, it's important for us not to confuse the eigenvalues λ_i of the adjacency matrix with the eigenvalues μ_i of the Laplacian matrix. There's only one specific case where it's easy to relate them – if G is d -regular, all vertices have the same degree. Then $L = dI - A$, so $\mu_i = d - \lambda_i$ for all i .*

Example 211

Next, let's derive Cayley's formula for the number of labeled trees on n vertices (equivalently, the number of spanning trees of K_n).

If G is the complete graph on n vertices, then the Laplacian matrix has all degrees $n - 1$ and adjacency matrix all non-diagonal entries 1, so

$$L = \begin{bmatrix} n-1 & -1 & \cdots & -1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{bmatrix}.$$

To find the number of spanning trees, we must compute $\det \tilde{L}$ – regardless of which principal cofactor we compute, we want the determinant of an $(n - 1) \times (n - 1)$ matrix with $(n - 1)$ s on the diagonal and -1 s everywhere else. The key observation is that all entries of $\tilde{L} - nI_{n-1}$ are -1 , so $\tilde{L} - nI_{n-1}$ has rank 1. Thus all eigenvalues of the matrix except one are equal to 0, and the trace of the matrix is $1 - n$, so the last eigenvalue is $1 - n$.

Therefore, if $\tilde{L} - nI_{n-1}$ has eigenvalues $0, 0, \dots, 0, -n + 1$, then the $(n - 1)$ eigenvalues of \tilde{L} (adding back nI_{n-1}) are $n, n, \dots, n, 1$. Thus the determinant of \tilde{L} , the product of the eigenvalues, is n^{n-2} , as desired.

In general, the matrix tree theorem doesn't tell us that there will be a nice formula for the number of spanning trees, but we actually get a nice closed form pretty often. To see one way in which this manifests, let G be a graph on n vertices without any multiple edges. We'll relate spanning trees of G to that of its **complementary graph** $\bar{G} = (V, \bar{E})$ (in which we include an edge between two vertices in \bar{G} if and only if there is no edge between those vertices in G).

To do that, define a new graph G^+ from a graph G by adding an extra vertex 0 (thinking of this as a root) which is connected to all of the other vertices of G . (Notice that G^+ is connected, even if G isn't.) Then assign a monomial weight to each spanning tree of G^+ to get a **graph polynomial**

$$F_G(x) = \sum_{T \text{ spanning tree of } G^+} x^{\deg_T(0)-1}.$$

(The x^n coefficient of F_G essentially tells us about how many connections to the root are needed to make T ; smaller exponents resemble spanning trees of G more.) Then the number of spanning trees of G is $\frac{F_G(0)}{n}$, because plugging in $x = 0$ into the graph polynomial only keeps the trees T where 0 is a leaf, and then there are n ways to attach 0 (we may notice that this has many similarities to our proof of the Cayley formula).

Theorem 212 (Reciprocity Theorem (Bedrosian, 1964))

If \bar{G} is the complementary graph to G (as defined above), then the graph polynomials satisfy

$$F_G(x) = (-1)^{n-1} F_{\bar{G}}(-x - n).$$

The proof is left as an exercise to us – there's a combinatorial proof, or we can express F_G in terms of the matrix tree theorem and show that two determinants are equal.

Example 213

Let's demonstrate the power of the reciprocity theorem by finding the number of spanning trees of the **complete graph** K_n (recovering Cayley's formula) and of the **complete bipartite graph** $K_{m,n}$ (a graph on $(m + n)$ vertices where there are mn edges between m vertices on one side and n vertices on the other, and no other edges).

- For the complete graph K_n , let O_n be the empty graph on n vertices. It has no spanning trees (or even edges), but O_n^+ has an extra vertex 0 connected to each of $1, 2, \dots, n$, so it has a single spanning tree which uses all n edges to the vertex 0. Thus, the graph polynomial for the empty graph is $F_{O_n}(x) = x^{n-1}$, and by the reciprocity theorem, that means that

$$F_{K_n}(x) = (-1)^{n-1} (-x - n)^{n-1} = (x + n)^{n-1}.$$

Plugging in $x = 0$, we indeed find that

$$\# \text{ spanning trees of } K_n = \frac{n^{n-1}}{n} = n^{n-2}.$$

- Next, for the complete bipartite graph, start with the disjoint union of two cliques $K_m \cup K_n$ (in which we have a set of m vertices all connected, and another set of n vertices all connected, with no edges between the two sets). To compute the graph polynomial of $K_m \cup K_n$, notice that once we add a vertex 0, we get a spanning tree of $(K_m \cup K_n)^+$ by picking a spanning tree of $(K_m)^+$ and also a spanning tree of $(K_n)^+$. Thus, the graph polynomials almost multiply directly, except that we only want to subtract off the -1 in the exponent once instead of twice:

$$F_{K_m \cup K_n}(x) = F_{K_m}(x) \cdot F_{K_n}(x) \cdot x = x(x + m)^{m-1}(x + n)^{n-1}.$$

The complement of this graph is the complete bipartite graph $K_{m,n}$, so by the reciprocity theorem (noting that $K_{m,n}$ and $K_m \cup K_n$ both have $m + n$ vertices)

$$F_{K_{m,n}}(x) = (-1)^{m+n-1}(-x - m - n)(-x - n)^{m-1}(-x - m)^{n-1} = (x + m + n)(x + n)^{m-1}(x + m)^{n-1}$$

Calculating the constant term $F_{K_{m,n}}(0)$ and dividing by the total number of vertices, we arrive at our result:

Proposition 214

The complete bipartite graph $K_{m,n}$ has $m^{n-1}n^{m-1}$ spanning trees.

As an exercise, we can try to prove this combinatorially as well. We'll discuss more examples next lecture!

27 April 17, 2019

We'll continue our discussion of the matrix tree theorem today. Recall that if we have a graph G with n vertices, we define the Laplacian matrix $L = (L_{ij})$ whose entries are

$$L_{ij} = \begin{cases} \deg_G(i) & i = j, \\ -\text{number of edges from } i \text{ to } j & i \neq j. \end{cases}$$

We can alternatively write $L = D - A$, where D is the diagonal matrix of degrees and A is the adjacency matrix of G . Last time, we defined the **reduced Laplacian** $\tilde{L} = L^i$, which is the Laplacian matrix without row and column i . Then the matrix-tree theorem says that the number of spanning trees in G is just the determinant of \tilde{L} .

We'll go through a proof of this theorem today, and we'll even generalize the result later. But first we'll discuss some more applications and examples:

Definition 215

Let $G = (V_1, E_1)$ and $H = (V_2, E_2)$ be two graphs. Define the **direct product** $G \times H$ of G and H to be the graph $(V_1 \times V_2, E_3)$, where edges are of the form

$$E_3 = \{(i, j), (i', j') : i = i', (j, j') \in E_2 \text{ or } j = j', (i, i') \in E_1\}.$$

For example, the product of two line graphs is a grid graph.

Lemma 216

Suppose the adjacency matrix $A(G)$ of G has eigenvalues $\{\alpha_1, \dots, \alpha_m\}$, and the adjacency matrix $A(H)$ of H has eigenvalues $\{\beta_1, \dots, \beta_n\}$. Then the adjacency matrix $A(G \times H)$ has eigenvalues $\{\alpha_i + \beta_j : 1 \leq i \leq m, 1 \leq j \leq n\}$.

This is an exercise in linear algebra – the idea is to notice that the adjacency matrix $A(G \times H)$ is a **tensor product**

$$A(G \times H) = A(G) \otimes I + I \otimes A(H),$$

so the tensor product of an eigenvector of $A(G)$ and $A(H)$ will be an eigenvector of $A(G \times H)$. It turns out this lemma doesn't help us very much with the matrix tree theorem except when our graphs are regular:

Proposition 217

Let G be a d_1 -regular graph with eigenvalues $\{\alpha_i\}$, and let H be a d_2 -regular graph with eigenvalues $\{\beta_j\}$. Then $G \times H$ is $(d_1 + d_2)$ -regular, and its eigenvalues are $\{d_1 + d_2 - \alpha_i - \beta_j\}$. Thus, number of spanning trees of $G \times H$ can be found by multiplying all eigenvalues (except the one at 0) and dividing by the number of vertices.

This allows us to find the number of spanning trees of some more complicated graphs:

Definition 218

The **hypercube graph** H_d is the 1-skeleton of a d -dimensional cube

In other words, we get H_d by taking the product of d copies of a 2-vertex chain (for example, the vertices of H_3 are the 8 vertices of a cube). Because H_d is a d -regular graph for any d , Proposition 217 applies to the hypercube graph. The adjacency matrix of a 2-vertex chain is

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \implies \lambda = \pm 1.$$

Thus, repeatedly applying Lemma 216, the adjacency matrix of the hypercube graph $A(H_d)$ has 2^d total eigenvalues of the form

$$\{\varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_d, \varepsilon_i \in \{-1, 1\}\}.$$

The smallest eigenvalue is $-d$, and the largest eigenvalue is d ; in general, the eigenvalue $-d + 2k$ shows up with multiplicity $\binom{d}{k}$, meaning that (because H_d is d -regular), the Laplacian has eigenvalue $2k$ of multiplicity $\binom{d}{k}$ (for any $0 \leq k \leq d$). Applying the matrix-tree theorem yields the following result:

Corollary 219

The hypercube graph H_d has

$$\frac{1}{2^d} \prod_{k=1}^d (2k)^{\binom{d}{k}} = 2^{2^d - d - 1} \prod_{k=1}^d k^{\binom{d}{k}}$$

total spanning trees.

For example, taking $d = 3$, the number of spanning trees for the skeleton of a cube is

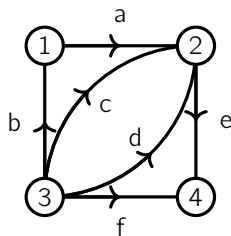
$$2^{8-3-1} \cdot 1^3 \cdot 2^3 \cdot 3^1 = 384.$$

As a sidenote, generalizing this result by taking the product of arbitrary chains (so that we have the skeleton of an arbitrary d -dimensional box instead of the hypercube) does not work very cleanly because our graph is no longer regular. But if we take the product of cycles instead (so that we have a torus), a similar product formula can be obtained.

Remark 220. *Some time ago, Richard Stanley proposed the problem of finding a combinatorial proof for Corollary 219. This is pretty difficult, but it was recently solved (around 2012) by Bernardi, though an explicit bijective construction still hasn't been found.*

In the rest of this lecture, we'll start on a proof of the matrix tree theorem.

Proof of Theorem 207. Our first step is to modify our adjacency matrix slightly. Direct each edge of the graph arbitrarily – for example, we can orient the edges of the graph in Example 203 as shown:



If G has n vertices and m edges, define the **oriented incidence matrix** of G to be the $n \times m$ matrix $B = (B_{ie})$, where (for any $i \in V$ and $e \in E$) we have

$$B_{ie} = \begin{cases} 1 & i \text{ is the source of edge } e, \\ -1 & i \text{ is the target of edge } e, \\ 0 & \text{otherwise.} \end{cases}$$

For example, the oriented adjacency matrix for the graph above looks like

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}.$$

Lemma 221

If B is the oriented adjacency matrix of G , then the Laplacian of G is $L = BB^T$.

Proof. By definition, $(BB^T)_{ij}$ is the dot product of the i th and j th rows of B . When $i = j$, we get a contribution of $(\pm 1)^2 = 1$ from each adjacent edge, which totals to the degree of the corresponding vertex. On the other hand, when $i \neq j$, we get $\pm 1 \cdot \mp 1 = -1$ each time an edge is directed from i to j or vice versa, so the off-diagonal entries indeed count the (negative) number of edges between i and j . (We can see now why the orientation of our edges is arbitrary.) \square

In particular, if \tilde{B} is the oriented incidence matrix with row i removed, the same argument also tells us that

$$\tilde{L} = \tilde{B}\tilde{B}^T.$$

To compute the determinant of the left-hand side (which is our goal), we can use the following result:

Theorem 222 (Cauchy-Binet formula)

Let A be a $k \times m$ matrix and B be an $m \times k$ matrix, where $k \leq m$. Then

$$\det(AB) = \sum_{\substack{S \subseteq \{1, 2, \dots, m\} \\ |S|=k}} \det(A^S) \det(B^S),$$

where A^S is the $k \times k$ submatrix of A with columns in S and B^S is the $k \times k$ submatrix of B with rows in S .

Proof. Consider the following block matrix product (which we can verify by direct computation):

$$\begin{bmatrix} I_k & A \\ 0 & I_m \end{bmatrix} \begin{bmatrix} A & 0 \\ -I_m & B \end{bmatrix} = \begin{bmatrix} 0 & AB \\ -I_m & B \end{bmatrix}.$$

Since all terms in this product are square matrices, we can take determinants of each term. The determinant of the first one is 1 (because it is an upper triangular matrix with diagonal entries 1), and the determinant of the right matrix is $\pm \det(AB)$. The claim is that the determinant of the middle matrix is exactly the right hand side of Cauchy-Binet up to that same \pm sign. We can illustrate this claim by example – for example, consider $k = 2, m = 4$, shown below:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 \\ -1 & 0 & 0 & 0 & b_{11} & b_{12} \\ 0 & -1 & 0 & 0 & b_{21} & b_{22} \\ 0 & 0 & -1 & 0 & b_{31} & b_{32} \\ 0 & 0 & 0 & -1 & b_{41} & b_{42} \end{bmatrix}.$$

Apply the usual definition of the determinant as a sum over permutations of products, and recall that permutations correspond to rook placements with a rook on each row and column. In particular, we should have one rook in each of the last $k = 2$ columns, after which we must pick the $m - k = 2$ -1 s in the bottom m rows that don't yet have rooks, and then we finish by placing rooks in the top $k = 2$ rows in the remaining columns. Another way to say this is that we pick some 2×2 submatrix of A and B , and in fact in order to get a nonzero contribution with the -1 s, we must pick the same set of indices S in A 's columns and B 's rows. Finally, we can check that the signs all “magically work out” (so that the sign on each product $\det(A_S) \det(B_S)$ is the same as the sign of $\pm \det AB$ in the matrix $\begin{bmatrix} 0 & AB \\ -I_m & B \end{bmatrix}$), which proves the result. \square

We're almost done, and we'll see how to use Cauchy-Binet to prove the matrix tree theorem next time. \square

28 April 19, 2019

Last lecture, we found that we could write the reduced Laplacian matrix of a graph G as $\tilde{L} = \tilde{B} \cdot \tilde{B}^T$, where \tilde{B} is the oriented incidence matrix of G with one row removed. We proved the Cauchy-Binet theorem last time, which allows us to write $\det(\tilde{L})$ as a sum involving \tilde{B} and \tilde{B}^T . If G has n vertices and m edges, then \tilde{B} is an $(n - 1) \times m$ matrix, and Cauchy-Binet tells us that

$$\det(\tilde{L}) = \det(\tilde{B} \cdot \tilde{B}^T) = \sum_{\substack{S \subseteq E \\ |S|=n-1}} (\det \tilde{B}^S)^2,$$

where we've used the fact that $\det A = \det A^T$ for a square matrix A . To finish the proof, we just need to show that this is indeed the number of spanning trees of G , and we can make the promising observation that subsets of the edges of size $n - 1$ can correspond to trees.

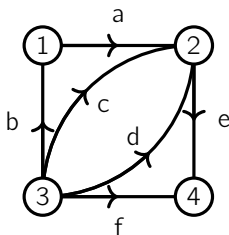
Lemma 223

Let $G = (V, E)$ be a graph, and let $S \subset E$ be a subset of the edges with $|S| = n - 1$. Then

$$\det \tilde{B}^S = \begin{cases} \pm 1 & S \text{ is the edges of a spanning tree,} \\ 0 & \text{otherwise.} \end{cases}$$

This lemma implies the matrix tree theorem, because each spanning tree contributes $(\pm 1)^2 = 1$ to the Cauchy-Binet sum, and all other subsets S give 0.

Proof. For illustration, consider the graph that we've been using as an example throughout these past few lectures:



Recall that the oriented incidence matrix (with columns a, b, c, d, e, f in that order) is

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}.$$

First, we show that if S is not the set of edges in a spanning tree, then $\det \tilde{B}^S = 0$. Indeed, notice that the column vectors in B corresponding to edges a, b , and c in our example are linearly dependent (namely, $\vec{a} + \vec{b} - \vec{c} = 0$), so the same is true about the corresponding column vectors in \tilde{B} (since we are just removing a row). Because the columns are linearly dependent, our matrix is singular and has determinant zero. More generally, if our subset of edges contains some cycle C of edges e_1, \dots, e_k , then there is some linear relation where

$$\pm \vec{e}_1 \pm \vec{e}_2 \cdots \pm \vec{e}_k = 0,$$

where the \pm s are $+$ if the edges agree with C 's orientation and $-$ otherwise. Since any subset S of $(n - 1)$ edges which is not a spanning tree contains some cycle, any such subset will have $\det \tilde{B}^S = 0$, as desired.

Now let's try to investigate the other case (where the edges of S form a spanning tree). We can check that if we remove any row of our example B and then take the determinant formed by columns a, c, f , we always get ± 1 . The way we'll explain this is that in a spanning tree, there is always some vertex which is a leaf, so the corresponding \tilde{B}^S matrix (as long as we don't remove that row) will only have one nonzero entry (which is ± 1).

Formally, let S be the set of edges that form a spanning tree. All trees on at least two vertices have at least two leaves, so pick any leaf ℓ such that ℓ is not the index of the removed row. Then the ℓ th row of the matrix \tilde{B}^S has only one nonzero entry, and it is a ± 1 . Expanding out the determinant along this row, we find that

$$\det(\tilde{B}^S) = \pm \det B',$$

where B' is an $(n - 2) \times (n - 2)$ matrix corresponding to a spanning tree on the remaining $(n - 1)$ vertices. Inductively, we can simplify this until we get to a tree on 2 vertices, which always has determinant ± 1 , so $\det(\tilde{B}^S) = \pm 1$, as

desired. □

With that, we've proved the matrix tree theorem, and we'll next think about some generalizations. First, we'll state a **weighted version** of the matrix tree theorem. Suppose that for every edge e of our graph G , we assign a weight x_e , and define the **weight of a spanning tree** to be

$$\text{wt}(T) = \prod_{e \text{ edge of } T} x_e.$$

Note that unlike the Cayley formula (which involves a product over vertex weights), this is a product over edge weights. Analogously to Cayley, though, we can define a polynomial

$$F_G = \sum_{T \text{ spanning tree}} \text{wt}(T),$$

and our goal is to determine this polynomial F_G .

Definition 224

Given a weighted graph $(G, \{x_e\})$, the **weighted Laplacian** $L = (L_{ij})$ of the graph is defined via

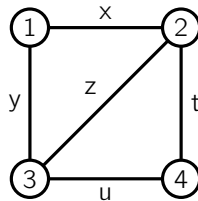
$$L_{ij} = \begin{cases} \sum_{e \text{ incident to vertex } i} x_e & i = j, \\ -x_e & e = (i, j), i \neq j, \\ 0 & \text{otherwise} \end{cases}$$

With this generalization, a similar result to the ordinary matrix tree theorem holds:

Theorem 225 (Weighted matrix tree theorem)

For any weighted graph G with weighted Laplacian L , we have $F_G = \det(\tilde{L})$, where \tilde{L} is the determinant of L with i th row and column removed for any i .

For example, consider the weighted version of the graph we've been using in the unweighted case, shown below:



Our weighted Laplacian matrix is then given by

$$L = \begin{bmatrix} x+y & -x & -y & 0 \\ -x & x+z+t & -z & -t \\ -y & -z & y+z+u & -u \\ 0 & -t & -u & u+t \end{bmatrix},$$

and the weighted matrix tree theorem says that **any** 3×3 minor gives us the sum of the weights of all spanning trees.

Proof. One way to think of a weighted graph is as a network, where edge weights are conductances. But for the purposes of our proof, notice that if the edge weights are **all nonnegative integers**, then a weight of k is equivalent

to having k edges between the relevant vertices (by linearity of the determinant). Thus in this case, the polynomial F_G is just the number of spanning trees of G^m , the graph where each weighted edge e is replaced by x_e unweighted copies of the edge.

Therefore, the matrix tree theorem implies the weighted matrix tree theorem whenever all edge weights are non-negative. But that means the more general result must also hold in all cases: we have two polynomials that agree on infinitely many points, so they must be the same polynomial. \square

Now that we've understood arbitrary edge weights, we'll add another layer of complexity by considering **directed graphs**. Let G be a **digraph** (directed graph) with no loops, but where multiple edges are allowed.

Definition 226

Let G be a **digraph** (directed graph), where multiple edges are allowed but not self-loops. An **out-tree** oriented at a vertex r is a spanning tree of G in the usual undirected sense, except that for every vertex i , all edges in the shortest path from r to i are directed away from the root. An **in-tree** has the same definition, except that all edges in the path from r to i must be directed towards the root instead.

Note that the number of out-trees and in-trees may not be equal (especially rooted at a given vertex r), but we claim that both of these counts can still be written as determinants, as long as we redefine our Laplacian matrix yet again:

Definition 227

For a directed graph G , the **outdegree** of a vertex v is the number of edges leaving v , and the **indegree** is the number of edges entering v . Define the matrices $L^{\text{in}} = (L^{\text{in}})_{ij}$ and $L^{\text{out}} = (L^{\text{out}})_{ij}$ via

$$(L^{\text{in}})_{ij} = \begin{cases} \text{indegree}(i) & i = j, \\ -\text{number of directed edges } i \rightarrow j & i \neq j, \end{cases}$$

$$(L^{\text{out}})_{ij} = \begin{cases} \text{outdegree}(i) & i = j, \\ -\text{number of directed edges } i \rightarrow j & i \neq j. \end{cases}$$

In other words, we can write $L^{\text{out}} = D^{\text{out}} - A$ and $L^{\text{in}} = D^{\text{in}} - A$, where A is the directed adjacency matrix. These matrices may not be symmetric anymore, but they still have some important properties – for example, the column sums of L^{in} are zero, but not necessarily the row sums, and the row sums of L^{out} are zero, but not necessarily the column sums.

Definition 228

For a square matrix L , define the **cofactor** $L^{ij} = (-1)^{ij} \det(L^*)$, where L^* is the matrix L with the i th row and j th column removed.

Theorem 229 (Directed matrix tree theorem)

For any $i \in \{1, 2, \dots, n\}$, the number of out-trees rooted at r is the cofactor $(L^{\text{in}})^{ir}$, and the number of in-trees rooted at r is $(L^{\text{out}})^{ri}$.

It turns out that even though this result is more general than our previous one, the proof is actually easier – we'll see it next time!

29 April 22, 2019

Last time, we stated the directed version of the matrix tree theorem, and we'll prove that result today. We'll review the setup today: given a directed graph G on n vertices (with no loops), $A = (a_{ij})$ is the adjacency matrix which encodes the number of edges from i to j (which is a matrix with nonnegative integer entries and zeros on the diagonal). Define two diagonal matrices D^{in} and D^{out} , which are the diagonal matrices with diagonal entries equal to the indegrees and outdegrees of the corresponding vertices, respectively. This gives us the two Laplacian matrices

$$L^{\text{in}} = D^{\text{in}} - A, \quad L^{\text{out}} = D^{\text{out}} - A.$$

We also defined the **cofactors** of a matrix L

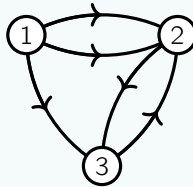
$$L^{ij} = (-1)^{i+j} \det(L \text{ without the } i\text{th row and } j\text{th column}).$$

These cofactors then tell us about the in-trees and out-trees of G : specifically, Theorem 229 states that the number of out-trees rooted at r is the cofactor $(L^{\text{in}})^{ir}$, and the number of in-trees rooted at r is $(L^{\text{out}})^{ri}$, for any $1 \leq i \leq n$.

Notice that the directed matrix tree theorem implies the undirected matrix tree theorem, since in the latter case in-degrees and out-degrees are equal, so $L^{\text{in}} = L^{\text{out}}$ are both the usual Laplacian matrix.

Example 230

Consider the directed graph G below:



The adjacency matrix of G is

$$A = \begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

and we have the Laplacian matrices

$$L^{\text{out}} = \begin{bmatrix} 3 & -2 & -1 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad L^{\text{in}} = \begin{bmatrix} 0 & -2 & -1 \\ 0 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

Now we can verify the directed matrix tree theorem for a few choices of r and i . First of all,

$$(L^{\text{out}})^{1,1} = \det \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 0,$$

which tells us that there are no in-trees rooted at 1 (indeed, no edges enter 1 at all). Similarly,

$$(L^{\text{out}})^{2,1} = -\det \begin{bmatrix} -2 & -1 \\ -1 & 1 \end{bmatrix} = 3,$$

and indeed there are 3 in-trees rooted at 2 in G (use the edge from 3 to 2 and one of the edges coming out from 1).

Finally,

$$(L^{\text{in}})^{2,1} = -\det \begin{bmatrix} -2 & -1 \\ -1 & 2 \end{bmatrix} = 5,$$

and indeed we can do the casework to see that there are 5 out-trees rooted at 1 in G .

It turns out the proof of the directed matrix tree theorem is easier than the undirected matrix tree theorem, because just like in the proof of Cayley's formula, having a strong induction hypothesis makes the argument more doable.

Proof of Theorem 229. The two statements (about in-trees and out-trees) map to each other by reversing all orientations of our graph edges, so it suffices to prove the statement about in-trees. In other words, let $\text{In}_r(G)$ be the number of in-trees of G rooted at r ; our goal is to show that this is equal to the cofactor $(L^{\text{out}})^{rk}$. We induct on the number of edges $i \rightarrow j$ of G where $i \neq r$ (that is, the number of edges not originating from the root).

The base case is where G is a graph where there are no edges out of non-root vertices; in this case, there are no in-trees because nothing goes into r . Then all entries of L^{out} that are not in the r th row are zero, so any cofactor $(L^{\text{out}})^{rk}$ is zero because the r th row is removed.

For the inductive step, pick any edge e from i to j , where $i \neq r$. Construct two graphs G_1 and G_2 , where G_1 is the graph G with edge e removed, and G_2 is G with all other edges (besides e) originating from i removed. Notice that any in-tree must use exactly one of the edges originating i (because all in-trees have root outdegree 0 and all other outdegrees 1), so

$$\text{In}_r(G) = \text{In}_r(G_1) + \text{In}_r(G_2).$$

There are now two cases. If there is only one edge originating from i (which is the edge $e : i \rightarrow j$ that we've picked out), we contract the vertices i and j . This preserves the number of in-trees (since i must connect to j in any in-tree), and we must check that the contraction also preserves the cofactor $(L^{\text{out}})^{rk}$. Indeed, the i th row of L^{out} only contains a 1 in the i th column and a -1 in the j th column, and contracting in this case means that we add the i th column to the j th column of L^{out} and then remove the i th row and column. But this contraction process (adding the two columns together) is the same as expanding the original determinant by minors along the i th row (which only contains a 1 and a -1) and combining the two resulting terms by swapping columns i and j in one of them. So in this case, both the number of in-trees and $(L^{\text{out}})^{rk}$ remain the same, and we've decreased the number of edges, so the result follows by induction.

In the other case, if we look at the Laplacian matrix, the i th row of $L^{\text{out}}(G)$ has some entries (a_1, \dots, a_n) . $L^{\text{out}}(G_1)$ looks almost identical to $L^{\text{out}}(G)$, except that one edge is removed, so that in row i we **decrease** a_i by one and **increase** a_j by one. On the other hand, $L^{\text{out}}(G_2)$ also looks identical to $L^{\text{out}}(G)$ except in row i , in which we have 0s except a 1 in the i th column and a -1 in the j th column. In particular, the sum of the i th rows of $L^{\text{out}}(G_1)$ and $L^{\text{out}}(G_2)$ add up to the i th row of $L^{\text{out}}(G)$. By linearity of determinants, this just means that **whenever we don't remove the i th row,**

$$(L^{\text{out}}(G))^{rk} = (L^{\text{out}}(G_1))^{rk} + (L^{\text{out}}(G_2))^{rk}.$$

Since the right hand side counts the number of in-trees for graph G_1 and G_2 separately (by induction, since G_1 and G_2 both have fewer edges than G), and $i \neq r$ by assumption, this proves the desired result, concluding the proof. \square

But often it's better to do a combinatorial proof – in this case, such a proof will help us understand why the direct matrix tree theorem looks at L^{out} for in-trees and vice versa.

Setup of the proof (which we'll do next time). Let's see another proof based on the involution principle – we'll only

show the result for the principal cofactors $k = r$, so we just wish to prove that

$$(L^{\text{out}})^{rr} = \sum_{\text{in-trees rooted at } r} \text{weight}(T).$$

Assign a weight x_{ij} to each pair of vertices (i, j) , so that our adjacency matrix contains off-diagonal entries x_{ij} . (Much like in the proof of the weighted matrix tree theorem, we may think of x_{ij} as the multiplicity of the directed edges from i to j .) As we should expect, the diagonal entries of our Laplacian matrix L^{out} are then $\sum_{j \neq i} x_{ij}$.

We'll do the full proof next time, but we'll illustrate the main point by example: for $n = 3$, the Laplacian matrix looks like

$$L^{\text{out}} = \begin{bmatrix} x_{12} + x_{13} & -x_{12} & -x_{13} \\ -x_{21} & x_{21} + x_{23} & -x_{23} \\ -x_{31} & -x_{32} & x_{31} + x_{32} \end{bmatrix}.$$

If we set $r = 3$, removing the last row and last column yields a cofactor of

$$(x_{12} + x_{13})(x_{21} + x_{23}) - x_{12}x_{21} = \boxed{x_{12}x_{23} + x_{13}x_{21} + x_{13}x_{23}}.$$

Notice that this algebraic expression encodes all ways that we can make an in-tree rooted at vertex 3 (where x_{ij} being included means that our in-tree contains the edge $i \rightarrow j$). The way we show this result in general is essentially to write the determinant as a sum over all permutations, getting collection of monomials that correspond to graphs. We can then construct a sign-reversing involution which preserves weights but changes $+$ signs to $-$ signs, so the only remaining terms are those coming from trees. But all of that will wait until next lecture! \square

In addition to having direct applications, the directed matrix tree theorem can also be used to prove algebraic identities. In particular, the following result is on our problem set:

Proposition 231 (Abel's identity)

For any real number z , we have

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} y(y + kz)^{k-1} (x - kz)^{n-k}.$$

In particular, this identity is the ordinary binomial theorem for $z = 0$.

30 April 24, 2019

We finished last class by discussing a "weighted directed matrix tree theorem," in which we assign a weight x_{ij} to each pair of indices $i \neq j$, defining a Laplacian matrix

$$(L^{\text{out}})_{ij} = \begin{cases} \sum_{k \neq i} x_{ik} & i = j, \\ -x_{ij} & i \neq j. \end{cases}$$

Our general result that we started formulating last time is as follows:

Theorem 232

The cofactor $(L^{\text{out}})^{rr}$ can be written as

$$(L^{\text{out}})^{rr} = \sum_{\substack{T \text{ in-tree} \\ \text{rooted at } r}} \text{weight}(T),$$

where the weight of an in-tree is the product of its edge weights x_{ij} (for each $i \rightarrow j$ in T).

By Cayley's formula, note that there are always n^{n-2} terms on the right hand side, because we can take any undirected tree on n vertices and then direct every edge towards the root.

Proof. Without loss of generality, let $r = n$. By definition of the determinant, the cofactor $(L^{\text{out}})^{nn}$ can be written as a sum over permutations

$$(L^{\text{out}})^{nn} = \sum_{w \in S_{n-1}} (-1)^{\ell(w)} \prod_{i=1}^{n-1} (L^{\text{out}})_{i w_i},$$

where $\ell(w)$ is the length of the permutation w , meaning that $(-1)^{\ell(w)}$ is the **sign** of w . Permutations break into cycles, so we'll write the product over the cycles (and fixed points, which correspond to diagonal entries): by the definition of the Laplacian matrix, we have an expression of the form

$$= \sum_{\substack{\text{subdivision of } [n-1] \\ \text{into disjoint cycles} \\ C_1 \cup \dots \cup C_m \cup \{\text{fixed points}\}}} \pm \left(\prod_{\substack{i \rightarrow j \text{ edge of} \\ \text{some } C_j}} x_{ij} \right) \cdot \left(\prod_{\substack{f \text{ fixed } j \neq f \\ \text{point}}} \sum x_{fj} \right),$$

To figure out the \pm sign for each term, notice that for a permutation σ with a single cycle of size c , we have $(-1)^{\ell(\sigma)} = (-1)^{c-1}$ (because $\ell(\sigma)$ is also the number of inversions). But when we write out our determinant as a polynomial in the x_{ij} s, we also get an additional contribution of $(-1)^c$ from each cycle, because the off-diagonal entries of our Laplacian matrix are $-x_{ij}$. This means that every cycle contributes a -1 , and therefore we get a factor of $(-1)^m$ for a term with m cycles:

$$(L^{\text{out}})^{nn} = \sum_{\substack{\text{subdivision of } [n-1] \\ \text{into disjoint cycles} \\ C_1 \cup \dots \cup C_m \cup \{\text{fixed points}\}}} (-1)^m \left(\prod_{\substack{i \rightarrow j \text{ edge of} \\ \text{some } C_j}} x_{ij} \right) \cdot \left(\prod_{\substack{f \text{ fixed } j \neq f \\ \text{point}}} \sum x_{fj} \right).$$

To interpret this combinatorially, we'll color the edges of our graph. For each subdivision of $[n-1]$, color the edges of the cycles red (but not the fixed points). Then to represent the $\sum_{j \neq f} x_{fj}$ term for each fixed point f , we pick an arbitrary green edge out from f into another vertex, which could potentially include vertex n . Thus, the expression above **sums over all directed graphs** H on the vertices $\{1, 2, \dots, n\}$ with $n-1$ edges colored in red or green, satisfying the following two conditions:

1. For all vertices $1 \leq i \leq n-1$, the outdegree is 1,
2. The red edges form a disjoint union of cycles.

Thus, we can rewrite our sum as

$$(L^{\text{out}})^{nn} = \sum_{\text{graphs } H} (-1)^{\text{number of red cycles}} \prod_{i \rightarrow j \text{ edge}} x_{ij},$$

which is starting to look more like what we want. The key point is that, in-trees rooted at n are exactly the digraphs without cycles with $n - 1$ edges, where the outdegree of every non-root vertex is 1 (that's condition (1) above). So the sum we're after is the sum over graphs H with **zero cycles** (and in particular zero red cycles); to show that the other contributions cancel out, we will now show a **weight-preserving, sign-reversing involution** ω on graphs H with at least one cycle.

But now that we know this is our goal, we can finish the proof quickly. We construct ω in the following way: fix a total ordering of edges $i \rightarrow j$ (e.g. lexicographically), so that we can also order the set of all cycles (also lexicographically). Then ω takes the first cycle in H and switch its color between green and red. To check that this is a valid involution, notice that the number of edges stays $(n - 1)$ and the outdegrees remain the same (because none of the edge positions have changed). Also, every cycle in H is always entirely red or entirely green because all outdegrees are 1 (meaning the path we can take from a given vertex is always determined), so we do not need to worry about violating condition (2) when swapping the color of a cycle, and this also shows that ω is its own inverse.

Because $\omega(H)$ and H have the same edges but different parities for the number of red cycles, their contributions always cancel out. Thus our sum for $(L^{\text{out}})^{nn}$ only leaves the contributions from in-trees, each of which has $(-1)^{\text{number of red cycles}} = (-1)^0 = 1$, concluding the proof. \square

With that, we're ready to move on to our next topic, **electrical networks**. This new topic is still related to the matrix tree theorem, but we'll start by discussing something we may have learned in physics class.

Given a graph G , we may think of its edges as **resistors** (which are the only circuit element that we'll use today). If we select two vertices A and B to be connected to some source of electricity, then a current will be generated through the edges. Our goal is to find the voltages and currents associated to the vertices and edges of G . We'll put this in a rigorous mathematical framework now:

Definition 233

Fix the orientation of edges in a graph $G = (V, E)$. Then for any edge e from vertices u to v , we have a **current** I_e through e , a **potential difference** or **voltage** V_e across e , and a positive real **resistance** R_e .

Note that the current through an edge is negated if we flip its orientation, so the actual choice is not too important. (If we were physicists, we'd talk about units for the current, voltage, and resistance, but we don't need them for mathematics.)

Theorem 234 (Ohm's law)

For any edge e , we have $V_e = I_e R_e$.

(This statement is really meant to define what resistance means, so it doesn't really warrant a proof.) Our main tools of reasoning about these electrical networks are the following two laws of physics:

Theorem 235 (First Kirchhoff's law)

The sum of the in-currents of any vertex v is the same as the sum of the out-currents:

$$\sum_{e:u \rightarrow v} I_e = \sum_{e':v \rightarrow w} I_{e'}$$

The first Kirchhoff's law is essentially a statement about **conservation of charge** (in the usual physical sense).

Theorem 236 (Second Kirchhoff's law)

The sum of the voltages along any cycle C of our graph is zero. Specifically, if C contains the edges e_1, \dots, e_m , then $\sum_{i=1}^m \pm V_{e_i} = 0$ (where we fix an orientation of C , and we have a $-$ exactly when an edge disagrees with that orientation).

Fact 237

The second Kirchhoff's law is equivalent to the existence of a **potential function** $U : \{\text{vertices}\} \rightarrow \mathbb{R}$ satisfying $V_e = U(v) - U(u)$ for any edge $e : u \rightarrow v$. In other words, suppose that we can "ground" one vertex by setting its voltage 0, and then we iteratively figure out the voltages of the whole graph by following edges arbitrarily. The second law then tells us that we will not run into any contradictions when we follow these edges, since two paths will always have the same voltage.

A typical problem in this framework is of the form "given all resistances of edges, find all of the currents and voltages." Doing so involves solving a system of linear equations – it turns out we can represent this system in terms of a Kirchhoff matrix, which is actually a Laplacian matrix with edge weights inversely proportional to resistances. But we'll talk about this in more detail next time!

31 April 26, 2019

Today, we will continue with "high school physics." To set up the problem more explicitly, let G be a connected digraph (the directions don't actually matter but they are used for convenience), and select two vertices A and B to be connected to a battery. Assume that we have some current I running through the battery, and there are two potential problems we might want to solve: (1) find all the relevant currents and voltages I_e, V_e for each edge e given the resistances R_e , or (2) find the resistance of the whole electrical network R_{AB} .

Last time, we mentioned three physics laws that govern the flow of electricity. We'll write them out as matrix laws now:

1. Kirchhoff's second law says that we can define a potential function U on the vertices of our graph such that

$$V_e = U_v - U_u$$

for any edge $e : u \rightarrow v$.

2. Ohm's law gives us a proportionality relationship between V_e and I_e , which we can write as

$$I_e = \frac{V_e}{R_e} = \frac{U_v - U_u}{R_e}.$$

3. Kirchhoff's first law states that for any vertex v , the in-current is equal to the out-current. For example, if $e_1 : v_1 \rightarrow v$ and $e_2 : v_2 \rightarrow v$ lead into v , and $e_3 : v \rightarrow v_3$, $e_4 : v \rightarrow v_4$, $e_5 : v \rightarrow v_5$ lead out of v , then

$$\frac{U_v - U_{v_1}}{R_1} + \frac{U_v - U_{v_2}}{R_2} = \frac{U_{v_3} - U_v}{R_3} + \frac{U_{v_4} - U_v}{R_4} + \frac{U_{v_5} - U_v}{R_5}.$$

Moving all terms to the left gives us a more symmetric expression of the form

$$\sum_{i \text{ neighbor}} \frac{U_v - U_{v_i}}{R_i} = 0.$$

But because the nodes A and B are connected to a battery, there is an additional current term that we must consider. Rearranging and taking that into account gives us our final relation:

$$\left(\frac{1}{R_1} + \dots + \frac{1}{R_d}\right) U_v - \sum_{i=1}^d \frac{1}{R_i} U_{v_i} = \begin{cases} 0 & v \neq A, B, \\ -I & v = A, \\ I & v = B. \end{cases}$$

Now we're ready to solve the problem: label the vertices of G from 1 to n , in particular setting $A = 1$ and $B = n$. If R_{ij} is the resistance of the edge connecting i and j , define the **conductance** of an edge

$$C_{ij} = \begin{cases} \frac{1}{R_{ij}} & (i, j) \text{ is an edge of our graph } G, \\ 0 & \text{otherwise.} \end{cases}$$

(The motivation for this definition is that resistances show up as reciprocals in our laws above.)

Definition 238

The **Kirchhoff matrix** K is an $n \times n$ matrix with entries

$$K_{ij} = \begin{cases} \sum_{\ell \neq i} c_{i\ell} & i = j, \\ -c_{ij} & i \neq j. \end{cases}$$

Notice that the Kirchhoff matrix is really just the Laplacian matrix for a graph whose edge-weights are conductances.

We can now write all three electrical laws in terms of this matrix! Specifically, if $\vec{U} = \begin{bmatrix} U_1 \\ \vdots \\ U_N \end{bmatrix}$ is the vector of potentials,

then we wish to solve the linear system of equations

$$K\vec{U} = \begin{bmatrix} -I \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix}.$$

Because all rows of the Kirchhoff matrix add to 0, this system has infinitely many solutions. But this makes sense, because our potential function can be shifted by a constant without changing the voltages and currents. More concretely, our Kirchhoff matrix K has rank $n - 1$ if G is connected (because we have at least one spanning tree, so all cofactors have to be nonzero by the matrix tree theorem).

Fact 239

More generally, if a graph G has k connected components, then its Laplacian matrix has rank $(n - k)$.

We'll now try to solve problem (2) with this linear algebra setup, meaning that we will try to find the **total effective resistance** $R_{AB}(G) = \frac{U_n - U_1}{I}$ explicitly. First of all, we can rescale \vec{U} and I without changing K , so we'll assume $I = 1$, and we'll also shift \vec{U} by a constant to set $U_1 = 0$ ("grounding the first vertex"). Then the resistance of the whole

circuit is $\frac{U_n - U_1}{I} = U_n$, and our linear system of equations simplifies to

$$K^{(1)} \begin{bmatrix} U_2 \\ \vdots \\ U_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix},$$

where $K^{(1)}$ denotes the $(n-1) \times (n-1)$ matrix obtained by removing the first row and column of K . This is a situation where we want to use **Cramer's rule** (despite Professor Strang saying to never do so in 18.06)! In particular, since we're trying to solve for $R_{AB}(G) = U_n$, Cramer's rule tells us that

$$R_{AB}(G) = U_n = \frac{\det \tilde{K}^{(1)}}{\det K^{(1)}},$$

where $\tilde{K}^{(1)}$ is obtained by replacing the last column of $K^{(1)}$ with the vector $\begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$ (the right-hand side of our matrix equation). But if we expand the determinant in the denominator along the last column, we just obtain the determinant of the upper left $(n-2) \times (n-2)$ square of $K^{(1)}$, which is the determinant of $K^{(1,n)}$ (that is, the Kirchhoff matrix with the first and last rows and column removed). With that, we've arrived at the following result:

Theorem 240

The overall resistance of our graph G is

$$R_{AB}(G) = \frac{\det K^{(1,n)}}{\det K^{(1)}} = \frac{\sum_{\tilde{T} \text{ spanning tree of } \tilde{G}} \text{weight}(\tilde{T})}{\sum_{T \text{ spanning tree of } G} \text{weight}(T)},$$

where the weight of a tree is the product of the conductances

$$\text{weight}(T) = \prod_{e \text{ edge of } T} \frac{1}{R_e},$$

and \tilde{G} is obtained from G by gluing vertices A and B together (or equivalently connecting the two with an edge of infinite conductance).

Another way to think about this numerator is that a spanning tree \tilde{T} of \tilde{G} is a forest of two components, with A and B in different components, so that when A and B are glued together, the two components also merge.

Example 241

Consider a graph G on three vertices A, B, C , with resistances of R_1 between A and C , R_2 between B and C , and R_3 between A and B .

Then Theorem 240 says that the total resistance between A and B is governed by spanning trees. To calculate the numerator, notice that \tilde{G} glues A and B together, so it only contains the edges R_1 and R_2 , either of which is a spanning tree. On the other hand, any two edges of G form a spanning tree of the original graph. Thus, we obtain the effective resistance

$$R_{AB}(G) = \frac{(R_1)^{-1} + (R_2)^{-1}}{(R_1 R_2)^{-1} + (R_1 R_3)^{-1} + (R_2 R_3)^{-1}}.$$

We could have also arrived at this result by using the concepts of **series and parallel connections** from physics:

Proposition 242 (Series connection)

Let G_1 and G_2 be two graphs which only share a single vertex C (the combined graph is denoted $G_1 + G_2$). Then $R(G_1 + G_2) = R(G_1) + R(G_2)$; in other words, for any $A \in G_1, B \in G_2$, we have $R_{AB} = R_{AC} + R_{BC}$.

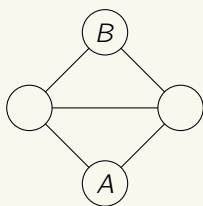
Proposition 243 (Parallel connection)

Let G_1 and G_2 be two graphs which share exactly two vertices A and B and no edges (the combined graph is denoted $G_1 \parallel G_2$). Then $R(G_1 \parallel G_2)^{-1} = R(G_1)^{-1} + R(G_2)^{-1}$ (where all resistances are taken between A and B).

Many graphs can be generated by combining smaller graphs with the $+$ and \parallel operations, but not all of them:

Fact 244

The smallest graph for which the effective resistance cannot be found directly with series and parallel connections is the **Wheatstone bridge**, shown below:



However, the general formula in Theorem 240 still works for this graph.

We'll now look at electrical networks with another perspective. This time, instead of rescaling our current to 1, rescale the current and shift the potential vector so that $U_A = 0$ and $U_B = 1$. Then all vertices have potentials between 0 and 1, and we can in fact view those potentials as probabilities:

Proposition 245

Consider a random walk on the vertices of G , where at each step, the probability of jumping to an adjacent edge is proportional to the conductance:

$$\Pr(u, v) = \frac{\frac{1}{R_{uv}}}{\sum_{w \neq u} \frac{1}{R_{uw}}}$$

Then the probability that a random walk starting at vertex v hits B before it hits A is U_v .

This result may remind us of the “drunk man problem” from our first lecture! In that setting, our graph G is a single line segment, A is the point $x = 0$, B is the house at $x = N$, and $U(x)$ is the probability that the man survives if starting at point x .

Proof sketch. If we let P_v be the probability of survival starting at vertex v , then P_v is a weighted average of the survival probabilities at the neighbors of v . Writing out the set of linear equations in matrix form, and also noting that $P_A = 0$ and $P_B = 1$, gives us exactly the same equations that we obtain with Kirchhoff’s law and the potential vector \vec{U} . □

That’s all we will say about electrical networks in this class, and we’ll continue with the **best theorem** next time.

32 April 29, 2019

Today we're going to talk about Eulerian cycles in graphs:

Definition 246

An **Eulerian cycle** (also an **Eulerian circuit**, **Eulerian tour**, or **Eulerian walk**) in a digraph G is a closed directed walk on G , where every edge is visited exactly once and the walk returns to its starting point.

We have the following famous result (historically, this is connected to the Seven Bridges of Königsberg problem, in which the goal was to cross each bridge exactly once):

Theorem 247 (Euler (1736))

A digraph G contains an Eulerian cycle if and only if G is connected as an undirected graph and all vertices of G have equal indegree and outdegree.

We can also say something similar for undirected graphs:

Corollary 248

An undirected graph G contains an (undirected) Eulerian cycle if and only if G is connected and all vertices have even degree.

Basically, if all vertices of an undirected graph have even degree, we can direct the edges in a way that satisfies Theorem 247.

Proof of Theorem 247. It is clear that the conditions are necessary, because an Eulerian cycle is connected, and we always enter and exit each vertex an equal number of times. To show that they are sufficient, we will construct an Eulerian cycle for the digraph. Start at an arbitrary vertex, pick some arbitrary edge out of it, and keep following arbitrary edges until we are stuck. Notice that for any vertex besides our starting one, if we enter it a times, we can exit it a times, so we can only get stuck at the starting vertex. At this point, we are either done, or there are some edges that we haven't yet used. In the latter case, there is some vertex for which an edge hasn't been used, so we can start the process from there again, repeating (finitely many times because we have finitely many edges) until our graph's edges are partitioned into a collection of cycles. Then because G is connected, we can combine those cycles together into a single Eulerian cycle (by repeatedly inserting one cycle into another). \square

We're interested in the next natural question, which is to **count the number of Eulerian cycles for a given digraph G** . To avoid overcounting, notice that every Eulerian cycle uses every edge once, we can fix a root r and an edge e out of r , and we'll require that all Eulerian cycles start from r and traverse e first.

Theorem 249 (B.E.S.T. Theorem (1951))

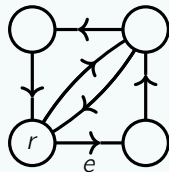
Let G be a connected digraph, where the indegree of each vertex is equal to its outdegree, and fix a root r . Let $\text{In}_r(G)$ be the number of in-trees rooted at r . Then

$$\text{number of Eulerian cycles in } G = \text{In}_r(G) \cdot \prod_{v \text{ vertex}} (\text{outdegree}(v) - 1)!$$

The acronym B.E.S.T. stands for the names “de **B**ruijn, von Aardenne-**E**hrenfest, **S**mith, and **T**utte.” Because we can calculate $\text{In}_r(G)$ using the directed matrix tree theorem, this theorem tells us that we can count the number of Eulerian cycles in terms of the Laplacian matrix of G .

Example 250

Let’s use the B.E.S.T. theorem to calculate the number of Eulerian cycles in the graph below:



We can verify that there are two in-trees rooted at r , and the outdegrees for the four vertices of the graph are 2, 1, 2, 1, so we should have $2 \cdot 1!0!1!0! = 2$ Eulerian cycles, which is indeed the case (there are two cases for which outgoing edge we take after the first two edges).

Proof of Theorem 249. Since our graph G is Eulerian (it satisfies the two requirements of Theorem 247), the two Laplacian matrices L^{in} and L^{out} are equal, and all row and column sums of our Laplacian matrix are zero. Thus, all cofactors of L are the same, and therefore the number of in-trees rooted at r is the same as the number of out-trees rooted at r , and that number is the same for all r .

Thus, it is indeed valid to fix a starting root r and starting edge e . For any Eulerian cycle C , list the edges of G in the order in which they appear, starting from e . This gives us a total ordering on the edges, denoted by $<_C$.

Now for any vertex $v \neq r$, mark the edge from v which is maximal (in $<_C$) among all outgoing edges – in other words, color the outgoing edge that occurs last in our Eulerian cycle. Let T be the subgraph formed by those colored edges. Notice that T is an in-tree rooted at r , because of the following observations:

- If G has n vertices, then T has $(n - 1)$ vertices.
- There is no fork in T (because there is only one outgoing edge from any vertex).
- There are no edges out of the root vertex r .
- **There are no directed cycles.**

This last claim requires more justification: suppose there were some cycle \tilde{C} in T . Because the root has no outgoing edges, \tilde{C} does not visit r ; let f be the maximal edge in \tilde{C} under the ordering $<_C$. We know that f is not the last edge in our Eulerian walk (because it doesn't go to the root), so let f' be the next edge in our **Eulerian** cycle C , and let f'' to be the edge that follows f in the cycle \tilde{C} . Because f is the maximal edge in \tilde{C} and $f <_C f'$, we have $f' \neq f''$. But $f'' <_C f$ by maximality of f , meaning that $f' > f''$. This is a contradiction, because f' and f'' originate from the same vertex of v , meaning we should have picked f' instead of f'' in our construction of T ! Thus no directed cycle \tilde{C} exists.

Therefore, our graph T indeed has outdegree 1 everywhere except at the root, which is enough to make it an in-tree. To finish the proof, for any vertex v , let w_v be the ordering of the edges outgoing from v , except for e and the edges in T , inherited from the order $<_C$. Notice that T and e together remove 1 from each outdegree, so each w_v corresponds to a permutation of size $\text{outdegree}(v) - 1$. We claim that specifying T and w_v tell us the entire ordering of edges, meaning that the map

$$C \text{ (Eulerian cycle)} \rightarrow (T, \{w_v\}_{v \text{ vertex}})$$

is a bijection! Indeed, to construct an Eulerian cycle from C , always pick the smallest edge from a vertex w_v when possible, only using a marked edge from T if required. This process will indeed give us an Eulerian cycle because we only use the last edge from a vertex when all other outgoing edges have already been traversed. \square

33 May 1, 2019

We'll move to another combinatorial object today and discuss **parking functions**. Here's the setup:

Problem 251

Suppose we have a one-way road with n cars. All n cars are trying to park, and there are n parking spots along the road. However, the drivers have favorite parking spots, encoded by a function $f : [n] \rightarrow [n]$ (meaning that the i th car prefers to park in the $f(i)$ th spot). The parking process goes as follows: each car (from 1 to n in order) tries to park in its favorite spot. If that spot is already taken, the car keeps driving and parks in the next available spot.

Example 252

If $(f(1), f(2), f(3), f(4)) = (2, 3, 1, 2)$, then the first car parks in spot 2, the second car parks in spot 3, the third parks in spot 1, and the fourth parks in spot 4. On the other hand, if $(f(1), f(2), f(3), f(4)) = (2, 3, 3, 2)$, the first car parks in spot 2, the second parks in spot 3, the third parks in spot 4, and the fourth car is out of luck.

Definition 253

A function $f : [n] \rightarrow [n]$ is a **parking function** if all n cars are able to park in the setup of Problem 251.

All permutations are parking functions (because there are no conflicts), but some other functions (like the first example of Example 252) work as well. The reason that the second example of Example 252 is not a parking function is that there are 4 cars that all want spots 2, 3, or 4, which is not satisfiable. It turns out this type of situation fully characterizes parking functions:

Lemma 254

A function $f : [n] \rightarrow [n]$ is a parking function if and only if for any $1 \leq k \leq n$, there are at most k values of i for which $f(i) \geq n + 1 - k$.

Intuitively, permutations are good parking functions, and it's also okay for us to decrease some entries of a permutation! For example, $(2, 3, 1, 2)$ is obtained by decreasing some entries of the permutation $(2, 3, 1, 4)$.

Lemma 255

A function $f : [n] \rightarrow [n]$ is a parking function if and only if there exists a permutation $w \in S_n$ such that $f(i) \leq w(i)$ for all i .

The proofs of these lemmas will be left as exercises for us. Putting that aside, let's try to find the number of parking functions as a function of n . One thing that makes our life easier is that Lemma 255 shows that the **order in which the cars arrive does not actually matter**. Looking at some small examples, there are 3 parking functions

for $n = 2$ $((1, 1), (1, 2), \text{ and } (2, 1))$. For $n = 3$, we look at all ways to decrease a permutation: we have 6 parking functions that are permutations of $(1, 2, 3)$, 3 of $(1, 2, 2)$, 3 of $(1, 1, 3)$, 3 of $(1, 1, 2)$, and 1 of $(1, 1, 1)$, for a total of 16. These numbers may remind us of Cayley's formula, and that's not a coincidence:

Theorem 256 (Pyke (1959), Konheim-Weiss (1966))

The number of parking functions $f : [n] \rightarrow [n]$ is $(n + 1)^{n-1}$.

One way to prove this result is to construct a bijection between spanning trees on $(n + 1)$ vertices and parking functions $f : [n] \rightarrow [n]$, but instead we'll show a short proof that doesn't use this strategy.

Proof. Modify our problem so that our road has $(n + 1)$ parking spots that are actually arranged in a **circle** (with cars driving counterclockwise around the circle.) The functions that dictate the cars' preferences is now a map $f : [n] \rightarrow [n + 1]$. Notice that in this setup, all cars will be able to park, and there is always one empty spot i .

We claim that the number of functions $f : [n] \rightarrow [n + 1]$ leading to the i th spot being empty is the same for all i . This is because the problem is entirely symmetric: if we have a function that leaves the i th spot empty, adding 1 to all values of $f \bmod (n + 1)$ yields a function that leaves the $(i + 1)$ th spot empty. But the functions f that result in spot $(n + 1)$ staying empty are exactly the parking functions, because $n + 1$ staying empty means both that no cars prefer spot $(n + 1)$, and no cars need to pass by the first n spots (because then they would take spot $(n + 1)$).

Thus, because there are $(n + 1)^n$ total functions from $[n]$ to $[n + 1]$, and a fraction $\frac{1}{n+1}$ of them are parking functions, we have $(n + 1)^{n-1}$ parking functions from $[n] \rightarrow [n]$, as desired. \square

This proof doesn't explain the connection between spanning trees and parking functions, but it does prove our result, which is good enough for now. Let's now go back to our original set-up and understand the implication of this theorem: if we pick a random uniform function $f : [n] \rightarrow [n]$, then the probability it is a parking function is

$$P_n = \frac{(n + 1)^{n-1}}{n^n} = \frac{1}{n} \left(1 + \frac{1}{n}\right)^{n-1}.$$

In the limit where we have many cars, notice that

$$\lim_{n \rightarrow \infty} P_n \approx \frac{1}{n} \cdot e = \frac{e}{n},$$

so the probability that all cars are able to park is proportional to $\frac{1}{n}$ for large n . ("Perhaps this explains why it is so hard to park in real life?")

Turning our attention now to **statistics on parking functions**, we'll first analyze the sum of the entries

$$s(f) = \sum_{i=1}^n f(i).$$

We know that f is always dominated by a permutation, so $s(f) \leq \binom{n+1}{2}$, with equality obtained for the $n!$ permutations. On the other hand, the minimum value $s(f) = n$ is only obtained by the single parking function $(1, 1, \dots, 1)$ – our question is to **compute the number of parking functions with some given sum**. It turns out that a natural way to interpret this statistic is in terms of trees:

Definition 257

Let T be a labeled tree on $(n+1)$ vertices, where we label the vertices $0, 1, \dots, n$. A pair (i, j) , where $1 \leq i < j \leq n$, is an **inversion** if j belongs to the shortest path from vertex i to the root 0 . Let $\text{inv}(T)$ denote the number of inversions of T .

Then we have an equidistribution-type result:

Theorem 258 (Kreweras (1980))

We have

$$\sum_{\substack{f: [n] \rightarrow [n] \\ \text{parking function}}} x^{\binom{n+1}{2} - s(f)} = \sum_{T \text{ tree on } \{0, \dots, n\}} x^{\text{inv}(T)}.$$

This polynomial is called the **inversion polynomial for trees** and is denoted $I_n(x)$.

For example, $I_n(1) = (n + 1)^{n-1}$, because each parking function contributes 1 to the sum on the left-hand side. Also, $I_n(0) = n!$ (where we set $0^0 = 1$), because we only count the $n!$ permutations which have maximal $s(f) = \binom{n+1}{2}$.

Remark 259. *The trees with no inversions are called **increasing trees**, and we see by equating constant coefficients that there are $n!$ of them. This is because an increasing tree can be constructed by connecting 1 to the root (since nothing can be between it and 0), then connecting 2 to either of the two existing vertices, then connecting 3 to one of the three existing vertices, and so on.*

There are some other interesting values that we can plug into Theorem 258 as well. If we set $x = -1$, then the right-hand side counts the difference between the number of trees with an even versus odd number of inversions. This turns out to be equal to the number of **alternating permutations** $w \in S_n$ satisfying $w_1 < w_2 > w_3 < \dots$, which are related to the Euler and Bernoulli numbers. We'll study this in more detail next time!

34 May 3, 2019

Our third problem set is now posted, and it will be due in a week. (Solving six problems completely is generally enough.) We'll discuss some concepts today that will be helpful for the problem set. Last time, we considered the tree inversion polynomial

$$I_n(x) = \sum_{\substack{T \text{ spanning} \\ \text{trees of } K_{n+1}}} x^{\text{inv}(T)} = \sum_{\substack{f: [n] \rightarrow [n] \\ \text{parking function}}} x^{\binom{n+1}{2} - \sum_i f(i)}.$$

Three of the problems from our problem set are related to the **alternating permutations** $w \in S_n$ satisfying the condition $w_1 < w_2 > w_3 \dots$.

Definition 260

Let A_n be the number of alternating permutations of S_n ; these are also known as **Euler**, **Andre**, and **zigzag numbers**, among other names.

These numbers show up in a variety of contexts:

Fact 261

Plugging in $x = -1$ into the tree-inversion polynomial yields $I_n(-1) = A_n$.

Fact 265

The Bernoulli numbers are related to the Riemann zeta function, which can be written in integral form as

$$\zeta(x) = \frac{1}{\Gamma(x)} \int_0^\infty \frac{u^{x-1}}{e^u - 1} du.$$

Using the definition of the Bernoulli numbers and plugging in integer values of x , we find that

$$B_{2n} = \frac{(-1)^{n-1} \cdot 2(2n)!}{(2\pi)^{2n}} \zeta(2n),$$

which allows us to find exact formulas for $\zeta(2)$, $\zeta(4)$, and so on.

Euler and Bernoulli numbers are difficult to obtain by direct series expansion or counting permutations by casework, but there is a recurrence relation that makes them more easy to calculate:

Proposition 266

The Euler numbers satisfy $A_0 = A_1 = 1$, and for all $n \geq 1$,

$$2A_{n+1} = \sum_{k=0}^n \binom{n}{k} A_k A_{n-k}.$$

In a sense, the Euler numbers are “labeled” Catalan numbers – if we remove the 2 and the $\binom{n}{k}$ coefficients, then we get the recurrence relation for C_n .

Proof. The base cases $A_0 = A_1 = 1$ are clear. For the recurrence relation, the left hand side counts the number of ways to have alternating permutations on $(n+1)$ elements that are either up-down or down-up: $w_1 < w_2 > w_3 \cdots$ or $w_1 > w_2 < w_3 \cdots$. We’ll show that the right hand side also counts the same quantity.

Indeed, for any alternating permutation w , the element $(n+1)$ must be in the $(k+1)$ th spot for some $0 \leq k \leq n$. Then the k elements before it, as well as the $(n-k)$ elements after it, must form some alternating sequence, and because $(n+1)$ is the largest element, the orientation (up-down or down-up) of both sequences are fixed. Thus, there are $\binom{n}{k}$ ways to pick which k elements appear before $(n+1)$, followed by $A_k \cdot A_{n-k}$ ways to arrange those k elements in alternating pattern and the other $(n-k)$ elements in another alternating pattern, and summing over all k gives us the desired recurrence relation. \square

It’s a good exercise for us to check that this recurrence relation can be rewritten as a differential equation for the exponential generating function $A(x)$:

$$2A'(x) = A^2(x) + 1.$$

In particular, we shift indices of an exponential generating function by taking a derivative (as opposed to dividing by x for the ordinary generating function). But other than that, this does still generally resemble the quadratic equation that we obtained for the Catalan number generating function.

To understand our next connection, we will need to discuss **binary trees**.

Definition 267

A **binary tree** is a rooted tree where each vertex has at most two children (where a vertex w is the **child** of a neighboring vertex v if v is on the shortest path from w to the root), known as the vertex's **left child** and **right child**. A **full binary tree** is a binary tree where every vertex either has no children (meaning that it is a leaf) or both a left and a right child.

Notice that a full binary tree can be constructed by repeatedly adding both a left and a right child to an existing leaf, so in particular all full binary trees have an odd number of vertices. It turns out both kinds of binary trees are related to Catalan numbers:

Proposition 268

The Catalan number C_n is the number of (ordinary) binary trees on n vertices, and it's also the number of full binary trees on $2n + 1$ vertices.

Proof. Constructing a bijection between these two sets is pretty straightforward. If we are given a binary tree on n vertices, we can add all missing children to the tree (so for each leaf, add both children, and for each vertex with only one child, add the other one). Since the original tree has n vertices, $(n - 1)$ of which serve as a child to another vertex, this process adds $2n - (n - 1) = n + 1$ vertices to our tree and gives us a full binary tree on $2n + 1$ vertices. For the reverse operation, removing all leaves from this tree gives us back the original tree, so these sets are indeed in bijection.

To show that both sets have C_n elements, we find a bijection between Dyck paths of length $2n$ and full binary trees with $2n + 1$ vertices. Given any Dyck path of length $2n$, we can break it up into sequences of consecutive up-steps, and we let a_1, a_2, \dots, a_k be the lengths of those groups. (Note that we can have $a_i = 0$ if there are two consecutive down-steps in the Dyck path; another way to say this is that a_i counts the number of up-steps on the line $y = x - 2n$.) We can then construct our full binary tree by creating a left chain of a_1 vertices (including all of their children), starting from the root. Next, we create a left chain of a_2 vertices starting from the bottom-most right child of the a_1 -chain, followed by a left chain of a_3 vertices starting from the bottom-most right child of the a_2 -chain, and so on. And in the special case where we have $a_i = 0$, we move one step up our tree and continue the process from there. This will always yield a valid full binary tree because the Dyck path property guarantees $\sum_{i=1}^k (a_i - 1) = 0$, and $a_i - 1$ is the number of left edges down the tree that we traverse when putting down a chain of a_i vertices. And we have a bijection, because we can recover the values of a_i by performing the same process (of depth-first search) and noting down the lengths of the various chains. \square

Next lecture, we'll discuss what happens when we label the vertices of our binary trees with positive integers. It will turn out that the number of increasing-labeled binary trees on n nodes is $n!$, while the number of increasing-labeled full binary trees on $2n + 1$ nodes is equal to A_n , which will explain our earlier thought about the Euler numbers being "labeled Catalan numbers."

35 May 6, 2019

Last lecture, we started discussing **binary trees**. Specifically, we found that the number of (unlabeled) binary trees on n vertices and the number of (unlabeled) full binary trees on $2n + 1$ vertices are both equal to C_n . Today, we'll formulate the labeled versions of those statements.

Theorem 269

A **labeled binary tree** is a binary tree where the n vertices are marked with the numbers 1 through n . Then the following are true:

1. The number of labeled binary trees on n vertices is $n!C_n$.
2. The number of **increasing** binary trees (for which the labels increase as we move away from the root) is $n!$.
3. The number of left-increasing binary trees (for which the left child's label is always larger than the parent's) is $(n+1)^{n-1}$.
4. The number of increasing full binary trees on $2n+1$ vertices is A_{2n+1} , the number of alternating permutations on $2n+1$ numbers.

Partial proof. The first point follows from Proposition 268 and the fact that we can label the n vertices arbitrarily in $n!$ ways. To prove the second and fourth points, we construct a bijection between increasing binary trees on n vertices and permutations in S_n . We've seen similar bijections in the past – we repeatedly “fold” an increasing binary tree by placing the root 1 in our permutation, placing everything on its left to the left of 1 in the permutation, placing everything on its right to the right of 1, and repeating the process with the roots of the sub-trees. To reverse the process, we find 1 and make it the root of our large tree, and for each of the left and right parts, find the smallest number in that subsequence and make those the roots of the smaller trees.

Since there are $n!$ permutations, there are also $n!$ increasing binary trees, as desired. And if our tree is full, we can verify by induction that we will always have a down-up permutation of size $2n+1$, of which there are the same number as up-down (alternating) permutations, A_{2n+1} . The last bijection and other details are left for us to work out ourselves. \square

We'll now switch topics and discuss **non-crossing paths**, which might help us with the remaining problems on the problem set. We know that if we have two vertices A and B in a square grid, where B is a steps above and b steps to the right of A , then the number of up-right lattice paths from A to B is $\binom{a+b}{a}$. Let's consider a more complicated setup:

Problem 270

Let A_1, A_2 be two starting points and B_1, B_2 be two target points in a square grid, where we want to connect A_1 to B_1 and A_2 to B_2 . Compute the number of ways to pick two such up-right **non-crossing** paths (which share no vertices).

We can actually write a formula to count such pairs of paths, which we'll now describe. Let G be an acyclic digraph (meaning that it has no directed loops) – for example, the one we're using in the problem above is the grid \mathbb{Z}^2 , with all edges directed up or to the right. Select vertices $A_1, \dots, A_k, B_1, \dots, B_k$ of G , and define $N(A_1, \dots, A_k, B_1, \dots, B_k)$ to be the number of ways to connect each A_i with its corresponding B_i using pairwise non-crossing paths, such that the paths do not share any vertices.

Theorem 271 (Lindstrom-Gessel-Viennot lemma, simple case)

With the setup above, suppose that $N(A_1, \dots, A_k, B_{w(1)}, \dots, B_{w(k)}) = 0$ unless w is the identity permutation. Then $N(A_1, \dots, A_k, B_1, \dots, B_k)$ is the determinant of the matrix $C = (C_{ij})_{i,j=1}^k$, where $C_{ij} = N(A_i, B_j)$.

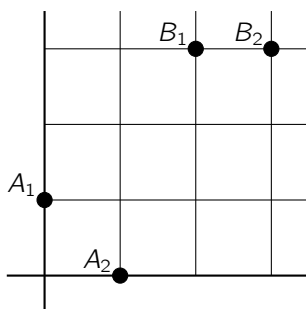
Fact 272

In other words, suppose that the only way to connect the set of A_i s and B_j s with noncrossing paths is if we match A_i with B_i . Then we can compute the total number of paths by taking a single determinant.

Example 273

Let $A_1 = (0, 1)$, $A_2 = (1, 0)$, $B_1 = (2, 3)$ and $B_2 = (3, 3)$. How many noncrossing paths are there from A_1 to B_1 and A_2 to B_2 ?

The diagram is shown below:



Notice that there is no way to draw up-right lattice paths from A_1 to B_2 and A_2 to B_1 without having the paths share at least one vertex. Thus, Theorem 271 applies, and we can compute $N(A_i, B_j)$ by evaluating the appropriate binomial coefficient. Our answer is then

$$\text{number of noncrossing paths} = \det \begin{bmatrix} \binom{4}{2} & \binom{5}{2} \\ \binom{4}{1} & \binom{5}{2} \end{bmatrix} = 20.$$

The more general result can be stated as well:

Theorem 274 (Lindstrom-Gessel-Viennot lemma, general case)

Suppose we have the same setup as Theorem 271 but without any assumptions on $N(A_1, \dots, A_k, B_{w(1)}, \dots, B_{w(k)})$. Then with the same definition of $C = (C_{ij})_{i,j=1}^k$ (where $C_{ij} = N(A_i, B_j)$), we have

$$\det C = \sum_{w \in S_n} (-1)^{\ell(w)} N(A_1, \dots, A_k, B_{w(1)}, \dots, B_{w(k)}).$$

We'll start the proof today and finish it next time:

Proof. We use the involution principle. Let $P(A_1, \dots, A_k, B_1, \dots, B_k)$ be the number of ways to connect the A_i s with B_j s using **any paths**; in particular,

$$P(A_1, \dots, A_k, B_1, \dots, B_k) = N(A_1, B_1) \cdot N(A_2, B_2) \cdots N(A_k, B_k),$$

where we recall that $N(A_i, B_j)$ counts the number of ways to get from A_i to B_j with **no restrictions**. By definition of the determinant,

$$\det C = \sum_{w \in S_k} (-1)^{\ell(w)} C_{1,w(1)} C_{2,w(2)} \cdots C_{k,w(k)} = \sum_{w \in S_k} (-1)^{\ell(w)} P(A_1, \dots, A_k, B_{w(1)}, \dots, B_{w(k)}).$$

This is the result we want but with P instead of N (meaning that we've written out all possible paths from the A s to the B s, without taking into consideration whether they cross or not). To get to the desired result, we must show that the contributions from crossing paths always cancel out.

The idea that we'll discuss in more detail next lecture is that our involution will find a crossing of two paths P_i and P_j , swapping them after that crossing (so that a section of P_j becomes part of P_i and vice versa). Such a crossing introduces an extra transposition (ij) to the permutation w , so the sign reverses and thus the contributions to $\det C$ cancel out. But we have to be careful in our choice of (i, j) to ensure that we do define an involution, and that's what we'll explain next time. \square

36 May 8, 2019

We'll start today's lecture by finishing the proof of the Lindstrom-Gessel-Viennot lemma proof (Theorem 274).

Proof of Theorem 274, continued. First, expand out $\det C$ as we did last time. We want to construct a sign-reversing involution σ on all "bad" collections of paths to show that their contributions cancel out. This involution σ is defined as follows: for any collection of paths where some of the paths share a vertex, find the "first intersection" (P_i, P_j, X) (where X is a vertex on both P_i and P_j), and swap P_i and P_j after the point X .

However, we can't just take the lexicographic minimum (i, j) to be the "first intersection," because that lexicographic minimum may not be preserved under σ . For example, consider the three paths below. if we swap paths $(1, 3)$ on the left, then the lexicographic minimum on the right becomes $(1, 2)$, so we would not recover the original collection of paths by applying σ again and would not get an involution.



Instead, we must be slightly more careful, and we pick (i, j) as follows: first, find the minimal index i such that P_i has an intersection with another path, and let the first intersection along that path (with another P_j) be point X . Among all of the paths that intersect P_i at X , pick the path P_j with smallest j . Under this choice, after applying σ , we'll still see the intersection of P_j and P_i at point X as our "first intersection." Since σ reverses the sign of our permutation w (as paths i and j 's final targets are swapped), the contribution of a configuration of paths \mathcal{P} and its swap $\sigma(\mathcal{P})$ contribute a total of zero to the sum $\det C$, showing the desired result. \square

This result also generalizes to a weighted version:

Theorem 275

Suppose all edges in our acyclic digraph G are assigned a positive weight $x_e > 0$, and define

$$N(A_1, \dots, A_k, B_1, \dots, B_k) = \sum_{\substack{\text{noncrossing paths} \\ P_1, P_2, \dots, P_k}} \prod_{i \in P_j} x_e.$$

Then defining a matrix $C = (C_{ij})_{i,j=1}^k$ like in Theorem 271 and Theorem 274, where $C_{ij} = N(A_i, B_j)$, we again have

$$\det C = \sum_{w \in S_n} (-1)^{\ell(w)} N(A_1, \dots, A_k, B_{w(1)}, \dots, B_{w(k)})$$

The same proof actually works again – we just need to observe that the sign-reversing involution σ also preserves the weights of the corresponding terms, because swapping the paths preserves the multiset of all edges.

In general, it is much easier to work with the “simple case” in Theorem 271 (where there is only one way to choose the targets and have non-crossing paths), so we’ll analyze that case a bit more.

Fact 276

Suppose G is a **planar graph** that can be drawn in a square, so that the source vertices A_1, \dots, A_k are on the left side (from bottom to top), and the target vertices B_1, \dots, B_k are on the right side (from bottom to top). Then the only way to draw non-crossing paths from the A_i s to the B_j s is by matching up each A_i with the corresponding B_i .

The matrix C that we obtain from such a setup, even in the weighted case, satisfies

$$\det C = N(A_1, \dots, A_n, B_1, \dots, B_n) \geq 0,$$

because we’re taking a sum of nonnegative-valued path weights. In fact, much more is true, as we’ll see in the following discussion.

Definition 277

A matrix C with real entries is **totally positive** (resp. **totally nonnegative**) if all minors (determinants of square submatrices) are positive (resp. nonnegative).

This condition is much stronger than having a matrix that is **positive definite**, which only requires that the **principal** minors are positive.

Example 278

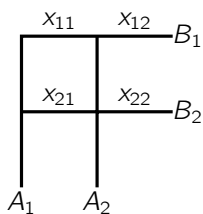
If we want a 3×3 matrix A to be totally positive, then all entries A_{ij} and $\det A$ must be positive, and we also have conditions like $A_{21}A_{33} > A_{31}A_{23}$ – there are many different conditions that must all be satisfied.

However, we do have one way to generate totally nonnegative matrices from the Lindstrom-Gessel-Viennot lemma. In particular, if we are in the setting of Fact 276, we can remove any set of k source vertices and k target vertices, and there will still be only one unique way to choose our path endpoints (meaning that any minor of C will be nonnegative), meaning that C is totally nonnegative. In fact, the converse is also true:

Fact 279

A matrix is totally nonnegative if and only if it can be represented as the matrix C in Theorem 275 for some planar graph as described in Fact 276.

In other words, there is a deep connection between the ideas of planarity and total nonnegativity, and we'll explore this more by explicitly explaining how to construct a general $m \times n$ totally positive matrix. Start by drawing an m by n grid, with all vertical edges directed upward and all horizontal edges directed to the right. It suffices to weight all vertical edges with 1s, and we place weights of x_{ij} on the horizontal edges coming out from the $m \times n$ grid. We place our A_i s from left to right on the bottom row, and we'll place our B_j s from top to bottom on the right column (by deforming the planar graph, this does indeed fall under the setting of Fact 276). For example, here is the setup for $m = n = 2$:

**Theorem 280**

With the setup above, construct a matrix C with entries $C_{ij} = N(A_i, B_j)$. Then any $m \times n$ totally positive matrix can be represented by choosing some positive real values for the x_{ij} s.

Corollary 281

The space of all totally positive $m \times n$ matrices is isomorphic to $(\mathbb{R}_{>0})^{mn}$.

We won't prove this result in lecture; instead, we'll now turn to another application of the Lindstrom-Gessel-Viennot lemma, studying **plane partitions**. Consider an $m \times n$ rectangle broken up into 1 by 1 grid squares, where we fill the grid with positive integers just like for Young tableaux, but where we are allowed to repeat numbers and our entries are required to be **weakly decreasing**. Here's an example of a plane partition:

7	7	6	6	4
7	6	5	4	3
5	5	3	3	3
3	3	3	2	1
2	2	2	1	1

One natural question is to count the number of plane partitions where each entry must lie in $\{1, \dots, k\}$. For example, the full list of plane partitions for $m = n = k = 2$ is shown below:

1	1	2	1	2	2	2	1	2	2	2	2
1	1	1	1	1	1	2	1	2	1	2	2

Proposition 282

The number of plane partitions for $k = 2$ is $\binom{n+m}{n}$, because the path that separates the 1s and 2s in the $m \times n$ square is an up-right lattice path.

Motivated by this simple case, if we want to consider the general- k case, we draw paths P_j for each $1 \leq j \leq k-1$, separating the entries larger than j from the entries at most j . This creates $(k-1)$ **weakly noncrossing paths** in an $m \times n$ grid, which are allowed to intersect but not cross over each other. We may now translate our paths so that they can no longer intersect – specifically, we have the path P_{k-1} starting at $A_{k-1} = (0, 0)$ and ending at $B_{k-1} = (m, n)$, the path P_{k-2} starting at $(1, -1)$ and ending at $(m+1, n-1)$, and so on. So our paths are non-crossing, and we can apply Lindstrom-Gessel-Viennot – we have a $(k-1)$ by $(k-1)$ matrix C with entries $C_{ij} = \binom{m+n}{m+i-j}$ (the number of ways to get from A_i to B_j), and the total number of plane partitions is $\det C$. There's actually an explicit product formula that we can extract, but we'll skip over it for now.

The shift in perspective is that we can think of our plane partitions as three-dimensional Young diagrams, where the entry c represents a tower of c cubes stacked on top of each other! Additionally, if we rotate those three-dimensional Young diagrams appropriately, we can biject plane partitions to **rhombus tilings** of an equiangular hexagon with side lengths m, n, k . This will tell us that there is a symmetry between m, n, k , and we'll look more at the explicit formula next lecture.

37 May 10, 2019

Last lecture, we started talking about plane partitions, which are $m \times n$ arrays of integers that are weakly decreasing across rows and columns. Let's now assume that the numbers that appear are in the set $\{0, 1, 2, \dots, k\}$ instead of $\{1, 2, \dots, k\}$, and let $P(m, n, k)$ denote the number of such plane partitions. We presented several ways to think about these plane partitions. First of all, we can represent plane partitions as a collection of k weakly noncrossing paths, which we can then translate to turn into noncrossing paths. The Lindstrom-Gessel-Viennot lemma then tells us that $P(m, n, k) = \det C$, where C is a $k \times k$ matrix with entries

$$C_{ij} = \begin{bmatrix} m+n \\ m+i-j \end{bmatrix}.$$

Here, we still have to calculate a determinant, but it's a bit of work.

Proposition 283 (MacMahon (1895))

The determinant of the matrix C above, and thus the number of plane partitions of an $m \times n$ box with all parts at most k , is

$$P(m, n, k) = \prod_{i=1}^m \prod_{j=1}^n \prod_{\ell=1}^k \frac{i+j+\ell-1}{i+j+\ell-2}.$$

Notice that this formula is symmetric in m, n, k , which may not be obvious from the determinant formula. For example, if $(m, n, k) = (5, 4, 3)$, then Lindstrom-Gessel-Viennot tells us that

$$P(5, 4, 3) = \det \begin{bmatrix} \binom{9}{5} & \binom{9}{4} \\ \binom{9}{6} & \binom{9}{5} \end{bmatrix} = \frac{2(9!)^2}{(5!)^2 \cdot 3!4!},$$

and we can check that this is indeed equal to the triple product obtained from Proposition 283.

Last class, we also mentioned that plane partitions can also be viewed as a set of cubes, where the labels on each grid square represent the number of stacked cubes (for example, $\begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix}$ is a $2 \times 2 \times 2$ cube with one cube removed).

This is essentially a "three-dimensional Young diagram," where we place a bunch of cubes in a $k \times m \times n$ box so that

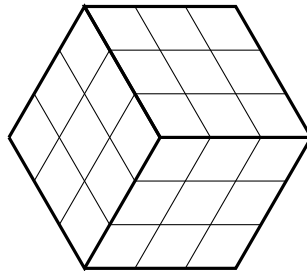
they are justified towards a specific corner. And by deforming the shape slightly, we can arrive at another combinatorial interpretation:

Proposition 284

$P(m, n, k)$ is the number of ways to tile an equiangular hexagon with sides of length m, n, k, m, n, k with tiles of the following shape (of side length 1):



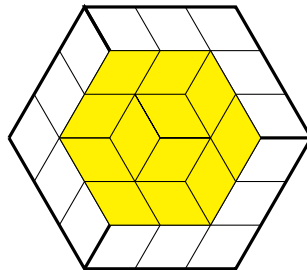
We'll explain this result by illustration. By default (if there are no cubes in the corresponding plane partition), our hexagon is tiled as shown:



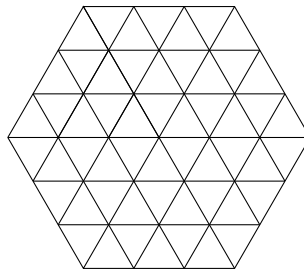
The center of the hexagon then serves as a "justified corner" for our cubes. For example, the plane partition

2	2	
2	1	

which we can think of as a collection of seven cubes inside the larger $3 \times 3 \times 3$ cube, yields the following rhombus tiling:



Yet another way to think of these rhombus tilings is to start with an equilateral triangle tiling of our hexagon, as shown below:



Rhombi are then formed by connecting two adjacent triangles with an edge. In other words, we consider the **planar dual graph** of our equilateral triangle grid (where dual vertices are faces of the original grid, and dual edges are drawn between adjacent faces), which is a hexagonal tiling. Then a rhombus tiling is a perfect matching of the vertices of the dual hexagonal tiling – this setting is also called a **dimer model** by physicists.

We'll return to rhombus tilings in the last lecture of the class, but for now we'll look at tilings of a similar grid:

Definition 285

A **domino tiling** of an $m \times n$ rectangular grid is a covering of the mn squares with 1×2 dominoes (with no holes). Let $N(m, n)$ denote the number of such domino tilings.

It is clear that if m and n are both odd, then $N(m, n) = 0$, because there are an odd number of squares but each domino takes up an even number of spots. Otherwise, we can color our $m \times n$ rectangle in the usual chessboard way, so that the black and white vertices form a bipartite graph. A domino tiling is then a perfect matching on this graph.

It turns out calculating $N(m, n)$ is related to computing the **permanent** of a matrix, which is like the determinant but without any negative signs in the sum over permutations. Unfortunately, permanents are much harder to calculate than determinants in general, because we don't have tools like eigenvalues to help us out! Kasteleyn figured out how to deal with this: instead of using the usual adjacency matrix, we replace some 1s with is .

Theorem 286 (Kasteleyn (1961))

Without loss of generality, let n be even. Then the number of ways to tile an $m \times n$ rectangle is

$$N(m, n) = \prod_{k=1}^{n/2} \prod_{\ell=1}^{\lfloor m/2 \rfloor} \left(4 \cos^2 \frac{\pi k}{n+1} + 4 \cos^2 \frac{\pi \ell}{m+1} \right),$$

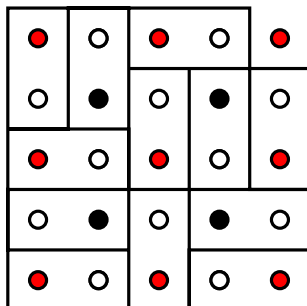
times another $(2 \cos \frac{\pi \lfloor m/2+1 \rfloor}{m+1})$ factor when m is odd.

This problem has applications to statistical physics, where computing the asymptotic number of matchings for large grids is important. While we're here, we'll also look at another domino-tiling result which can help us get such approximate asymptotics:

Theorem 287 (Temperley (1974))

Suppose $m = 2k + 1$ and $n = 2\ell + 1$ are both odd. Then the number of ways to domino tile an $m \times n$ grid without a corner square is the number of spanning trees of a $k \times \ell$ grid.

Proof. We will use **Temperley's bijection**. Given a domino tiling, color the underlying $(2k + 1)$ by $(2\ell + 1)$ grid black and white so that the corners are black, and then take the black entries (i, j) where i, j are both odd and color them red. We show $m = n = 5$ below for illustration:



Now we can form a spanning tree on the red $k \times \ell$ grid in the following way: if there is a domino that points along the line connecting two adjacent red nodes, then draw a line connecting them. Showing that this always yields a spanning tree (and that every spanning tree corresponds to a domino tiling) is left as an exercise for us.

Solution by Fadi Atieh. To show that (1) implies (2), we prove the contrapositive – if there are more than k cars that want to park in the last k spots (corresponding to having parking function values at least $n - k + 1$), then by Pigeonhole there will not be locations for all of the cars to park. Similarly, to show that (2) implies (1), again prove the contrapositive. Assume that f is not a parking function, so there exists some first car i that is unable to park. This means that the $n - f(i) + 1$ parking spots in the set $S = \{f(i), f(i) + 1, \dots, n\}$ are all occupied. If all of these parking spots are occupied by previous cars that wanted to park in spots $f(i)$ or higher, then we have proved (2) false with the value $k = f(i)$ (because we have the cars that are parked in the spots S , plus car i). Otherwise, the parking spot $f(i) - 1$ must have also been occupied (so that some car was pushed from numbers below $f(i)$ to spot $f(i)$ or higher), so repeat the argument by adding $f(i) - 1$ to S . This process must stop at some point because we have finitely many cars, and the only possibility is that some value of k proves (2) false.

The rest of the proof is simpler: to show that (1) implies (3), take a parking function f and construct the permutation where car i goes to its eventual parking spot $w(i)$. Since a car's parking spot is at least as large as its favorite spot, we indeed have $f(i) \leq w(i)$ for all i . And to show that (3) implies (1), we start with the parking function given by $w(i)$ (which satisfies (2)), and we decrease entries one by one. Making entries of a parking function smaller can never cause us to violate condition (2), so the eventual function f is indeed a parking function, concluding the proof. \square

Problem 290

Give a proof of Abel's identity (Proposition 231).

Solution by Congyue Deng. Consider a weighted digraph on $(n+2)$ vertices labeled A, B , and $\{1, 2, \dots, n\}$, where we have an edge from A to B of weight 1, from A to each of $\{1, 2, \dots, n\}$ with weight x , from B to each of $\{1, 2, \dots, n\}$ with weight y , and between any two elements of $\{1, 2, \dots, n\}$ of weight z (in both directions). We will calculate the total weight of out-trees rooted at vertex A .

First, we'll calculate the contributions directly. All out-trees rooted at A must include the edge $A \rightarrow B$, and then k of the vertices of $[n]$ have paths that come from A without going through B , while the other $n - k$ vertices lie on paths that do pass through B . Because the edges to the two groups of vertices can be independently chosen, we can calculate the weight contributions separately and multiply them together. If we want to connect a given set of k vertices to A , we can form a forest of $i + 1$ connected components in the complete graph K_k , each of which must be connected to an edge originating from A . There are $\binom{k-1}{i} k^{k-1-i}$ such configurations ($\binom{k}{i+1}$ ways to choose the vertices receiving outgoing edges from A , and then $(i+1)k^{k-i-2}$ ways to form $(i+1)$ components on K_k with those vertices all in distinct components – see [this stackexchange post](#), where k is replaced with $k - i$ and n is replaced with $k + 1$), and because each of these configurations uses $i + 1$ edges to A and $k - i - 1$ edges between the elements of $[n]$, summing over all possible i yields

$$\sum_{i=0}^{k-1} \binom{k-1}{i} k^{k-1-i} x^{i+1} z^{k-i-1} = x(x + kz)^{k-1}.$$

Similarly, the other $n - k$ vertices can be connected with a total weight contribution of $y(y + (n - k)z)^{n-k-1}$. Thus, the total sum of weights across all out-trees of our digraph rooted at A is

$$\sum_{k=0}^n \binom{n}{k} x(x + kz)^{k-1} y(y + (n - k)z)^{n-k-1}.$$

But on the other hand, we can use the directed matrix tree theorem to compute the total sum of out-tree weights. Letting the rows correspond to $A, B, 1, \dots, n$, we find that the Laplacian has first row $(1 + nx, -1, -x, \dots, -x)$,

second row $(0, ny, -y, \dots, -y)$, bottom $n \times n$ matrix having diagonal entries $x + y + (n - 1)z$ and all other entries $-z$, and all other entries zero. Taking the first principal cofactor, our determinant is $(x + y)(x + y + nz)^{n-1}$ (by taking the product of eigenvalues and using a similar strategy as we did when rederiving Cayley's formula). This yields the identity

$$(x + y)(x + y + nz)^{n-1} = \sum_{k=0}^n \binom{n}{k} x(x + kz)^{k-1} y(y + (n - k)z)^{n-k-1}.$$

If we then replace y with $y - nz$, the result follows by induction on n . \square

Alternative solution by Mathew Ganatra. We'll induct on n directly. The base case $n = 0$ is clear, and the case $n = 1$ can also be directly verified:

$$\sum_{k=0}^1 \binom{1}{k} y(y + kz)^{k-1} (x - kz)^{1-k} = x + y = (x + y)^1.$$

Now for the inductive step, assume that we know (from the inductive hypothesis) that

$$(x + y)^{n-1} = \sum_{k=0}^{n-1} \binom{n-1}{k} y(y + kz)^{k-1} (x - kz)^{n-k-1}.$$

We can show the result for $(x + y)^n$ by checking that the x - and y -derivatives of both sides of Abel's identity are equal. Indeed, the left-hand side has

$$\frac{d}{dx}((x + y)^n) = n(x + y)^{n-1}, \quad \frac{d}{dy}((x + y)^n) = n(x + y)^{n-1}$$

while the right-hand side has

$$\frac{d}{dx} \left[\sum_{k=0}^n \binom{n}{k} y(y + kz)^{k-1} (x - kz)^{n-k} \right] = \sum_{k=0}^n \binom{n}{k} y(y + kz)^{k-1} (n - k)(x - kz)^{n-k-1}$$

(where we've used the product rule). Since $\binom{n}{k}(n - k) = n\binom{n-1}{k}$, this now simplifies (by the induction hypothesis) to

$$n \sum_{k=0}^{n-1} \binom{n-1}{k} y(y + kz)^{k-1} (x - kz)^{n-k-1} = n(x + y)^{n-1},$$

which matches with the left-hand side. Meanwhile, by the product rule,

$$\begin{aligned} \frac{d}{dy} \left[\sum_{k=0}^n \binom{n}{k} y(y + kz)^{k-1} (x - kz)^{n-k} \right] &= \sum_{k=0}^n \binom{n}{k} ((k - 1)y + y)(y + kz)^{k-2} (x - kz)^{n-k} \\ &= n \sum_{k=0}^n \binom{n-1}{k-1} y(y + kz)^{k-2} (x - kz)^{n-k} \end{aligned}$$

because $\binom{n}{k}k = n\binom{n-1}{k-1}$. Shifting indices here again ($k \mapsto k + 1$) allows us to use the inductive hypothesis and find that this is also equal to $n(x + y)^{n-1}$. Thus, the derivatives of the two sides of Abel's identity are indeed equal, and it remains to check that the constant terms are also equal. Indeed, if we set $x = -y$, the left side becomes 0, and the the right hand side also vanishes because we have alternating binomial coefficients. This concludes induction and proof. \square

Problem 291

Compute the number of Eulerian cycles in a bidirected n -cube graph.

Solution by Sophia Xia. By the B.E.S.T. theorem, we want to compute the number of intrees rooted at some vertex r , multiplied by $(\text{outdeg}(v) - 1)!$ for all v . An n -cube has 2^n vertices, each with outdegree n , so the vertex contribution is $\boxed{((n-1)!)^{2^n}}$.

We also know that the number of spanning trees on an n -cube is equal to the number of in-trees, because we can direct all edges of a spanning tree towards the root. Corollary 219 gives us a factor of $\boxed{2^{2^n - n - 1} \prod_{k=1}^n k^{\binom{n}{k}}}$, and multiplying these together yields our answer. \square

Problem 292

Prove that the exponential generating function for the Euler numbers is

$$\sum \frac{A_n}{n!} x^n = \tan x + \sec x.$$

Solution by Song Wenzhu. We rewrite the recurrence relation (from lecture)

$$2A_{n+1} = \sum_{k=0}^n \binom{n}{k} A_k A_{n-k}$$

in the more suggestive form

$$2(n+1) \frac{A_{n+1}}{(n+1)!} = \sum_{k=0}^n \frac{A_k}{k!} \frac{A_{n-k}}{(n-k)!}.$$

Define $c_k = \frac{A_k}{k!}$, and notice that the exponential generating function $G(x) = \sum_k c_k x^k$ satisfies

$$G(x)^2 = c_0^2 + \sum_{k=0}^1 (c_k c_{1-k})x + \cdots + \sum_{k=0}^n (c_k c_{n-k})x^n.$$

Plugging in our rewritten recurrence relation, this simplifies to

$$= c_0^2 + \sum_{n=2}^{\infty} 2n c_n x^{n-1} = 2G'(x) - 1$$

(since $c_0^2 = 1$, but if we had included $n = 1$ in our sum we would have a constant term of 2 instead). This is a differential equation:

$$\frac{G'}{G^2 + 1} = \frac{1}{2} \implies \arctan G = \frac{1}{2}x + C,$$

and the initial condition $G(0) = 1$ yields

$$G = \tan\left(\frac{1}{2}x + \frac{\pi}{4}\right) = \tan x + \sec x$$

(after some verification with trig identities). \square

Problem 293

Find bijections between the following sets:

1. the set of labeled trees on $n + 1$ vertices,
2. the set of plane binary trees on n vertices labeled by $[n]$ such that the left child of a vertex always has a bigger label than its parent,
3. the set of Dyck paths of length $2n$ with up steps labeled by $[n]$ (and unlabeled down steps) such that, for any two consecutive up steps, the label of the second step is greater than the label of the first step,
4. the set of parking functions of size n .

Proof by Vanshika Jain. We'll do just one of the bijections for time (between (2) and (3)). Given a plane binary tree, perform a "depth-first search," where we traverse the tree by taking the leftmost edges, turning back around whenever we run out of space. Then every time we move left on the tree, we go up on the Dyck path, and every time we move right, we go down on the Dyck path. (Additionally, start the Dyck path with an up step and end it with a down step.) The i th up step in the Dyck path then lands on the vertex labeled i in our tree.

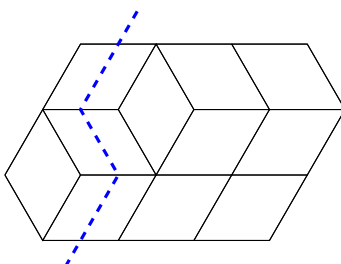
This bijection works because consecutive sequences of up steps are always increasing (and correspond to the left child always has a bigger label), and we never go below the x -axis in our Dyck path because the tree depth is equal to the Dyck path height. For the inverse map, we just look at our up and down steps and travel in the corresponding direction when making our tree (after an up step, another up step means a left child, a down and then up step means a right child, and more down steps mean that we retrace our tree.) \square

39 May 15, 2019

In our last lecture, we'll talk some more about **plane partitions**. Recall that plane partitions can be thought of as a grid of $m \times n$ numbers in the range $\{0, k\}$ that are weakly decreasing downward and to the right. By Lindstrom-Gessel-Viennot, the number of such plane partitions is the determinant of the $k \times k$ matrix C whose entries are $C_{ij} = \binom{m+n}{m+i-j}$, and MacMahon derived that this determinant is the product $\prod_{i=1}^m \prod_{j=1}^n \prod_{\ell=1}^k \frac{i+j+\ell-1}{i+j+\ell-2}$.

But there are also several geometric ways to think about plane partitions – they correspond to rhombus tilings of equiangular hexagons with side lengths m, n, k, m, n, k , as well as perfect matchings in a honeycomb graph. Today, we'll describe yet another interpretation: plane partitions can be viewed as **pseudo-line arrangements**.

Pseudo-lines are like lines, but they don't have to be straight. Here's the process for converting a rhombus tiling into a pseudo-line: for each pair of opposite sides of our hexagon, draw pseudo-lines by following the common edges of rhombi with parallel sides. For example, pseudo-lines from top to bottom must start from a rhombus with a horizontal edge along the top edge of the hexagon, and they must connect this rhombus to the neighboring rhombus on its bottom edge. (We then repeat until we reach the bottom edge of our hexagon.) Here's an example of a pseudo-line:



Because the set of all pseudo-lines (across all three pairs of sides of our hexagon) intersect once per rhombus, the configurations of psuedo-lines that we draw are in one-to-one correspondence with our rhombus tilings. And if we study this topic a little more, we'll see connections to the **theory of symmetric polynomials**, which are related to Young diagrams and the Schensted correspondence, among many other things we've discussed in this class. We'll explore this a bit now:

Definition 294

A polynomial $f(x_1, \dots, x_n)$ is **symmetric** if it is invariant under the permutations of the variables (in other words, $f(x_1, \dots, x_n) = f(x_{w(1)}, \dots, x_{w(n)})$ for all permutations $w \in S_n$).

The simplest polynomials are the **elementary symmetric polynomials** e_1, \dots, e_n , where x_k is the sum of all square-free monomials in the variables $\{x_1, \dots, x_n\}$ of degree k . For example,

$$e_1 = x_1 + \dots + x_n, \quad e_2 = x_1x_2 + x_1x_3 + \dots + x_{n-1}x_n.$$

These polynomials are important because of the following algebraic fact:

Theorem 295

Any symmetric polynomial $f(x_1, \dots, x_n)$ can be written uniquely as a polynomial in the elementary symmetric polynomials e_1, \dots, e_n .

To show why this is useful, we'll introduce a new combinatorial object:

Definition 296

A **semi-standard Young tableau** (SSYT) for a Young diagram λ is a filling of λ with the numbers $1, 2, \dots$, such that the entries increase **strictly** in columns and **weakly** in rows.

While standard Young tableaux require us to use each number exactly once, there is no such requirement for semi-standard Young tableaux. Below is an example:

1	1	1	1	2	2	2	3
2	2	2	3	3	4	4	
3	3	3	4	4			
4	4	6	6				

Definition 297

Let λ be a Young diagram with at most n rows. The **Schur polynomial** associated to λ is

$$s_\lambda(x_1, \dots, x_n) = \sum_{\substack{\text{SSYT of} \\ \text{shape } \lambda \\ \text{filled with} \\ \{1, 2, \dots, n\}}} x_1^{\#1\text{'s}} \dots x_n^{\#n\text{'s}}$$

Notice that λ cannot have more than n rows, because the first entry of row i is always at least i .

Example 298

The Schur polynomial for a single box is just $x_1 + x_2 + \cdots + x_n = e_1$. More generally, the Schur polynomial for a column of k boxes is the k th symmetric polynomial, since we pick some k of the n numbers to put in our semi-standard Young tableau (and place them in sorted order).

Example 299

Because adjacent entries can be equal in rows of a semi-standard Young tableau, the Schur polynomial for a horizontal domino is $\sum_{i < j} x_i x_j = x_1^2 + x_1 x_2 + x_2^2 + \cdots = e_1^2 - e_2$.

Theorem 300

Any symmetric polynomial $f(x_1, \dots, x_n)$ can be written uniquely as a linear combination of Schur polynomials

$$f(x_1, \dots, x_n) = \sum_{\lambda} c_{\lambda} s_{\lambda}(x_1, \dots, x_n).$$

In other words, Theorem 300 tells us that the Schur polynomials form a **linear basis**, while Theorem 295 tells us that the elementary polynomials form an **algebraic basis**. We can actually relate the two results by writing the Schur polynomials in terms of the elementary polynomials:

Theorem 301 (Jacobi-Trudi)

Let $\lambda = (\lambda_1, \dots, \lambda_e)$ be a partition, where $e \leq n$, and let $\lambda' = (\lambda'_1, \dots, \lambda'_m)$ be its **conjugate partition** (recall that this means we take the “transpose Young diagram”). Then

$$s_{\lambda}(x_1, \dots, x_n) = \det \begin{bmatrix} e_{\lambda'_1} & e_{\lambda'_1+1} & \cdots & e_{\lambda'_1+(m-1)} \\ e_{\lambda'_2-1} & e_{\lambda'_2} & \cdots & e_{\lambda'_2+(m-2)} \\ \vdots & \vdots & \ddots & \vdots \\ e_{\lambda'_m-(m-1)} & e_{\lambda'_m-(m-2)} & \cdots & e_{\lambda'_m} \end{bmatrix},$$

where we assume $e_0 = 1$ and $e_{-k} = 0$ for all $k < 0$.

For example, the Schur polynomial of the horizontal domino, where $\lambda = (2)$ and $\lambda' = (1, 1)$, is indeed

$$s_{(2)} = \det \begin{bmatrix} e_1 & e_2 \\ e_0 & e_1 \end{bmatrix} = e_1^2 - e_2.$$

Fact 302

The form of this determinant may remind us of the determinant of binomial coefficients we obtained when counting the number of plane partitions with Lindstrom-Gessel-Viennot, and in fact it’s possible to connect these ideas. Specifically, if we think of semi-standard Young tableaux using noncrossing lattice paths, we can indeed prove Jacobi-Trudi in this way (for example, see this [paper](#)).

Because the Schur polynomial is a sum over semi-standard Young tableaux, we can plug in $x_1 = \cdots = x_n = 1$ into Jacobi-Trudi, which yields the following result:

Theorem 303

The number of semi-standard Young tableaux of shape λ filled with numbers in $\{1, 2, \dots, n\}$ is $\det C$, where $C = (C_{ij})_{i,j=1}^{|\lambda|}$ has entries

$$C_{ij} = \binom{n}{\lambda'_i + j - i}.$$

It turns out that there are two ways to write this determinant as a product:

$$\det C = \prod_{1 \leq i < j \leq n} \frac{\lambda_i - \lambda_j + j - i}{j - i} = \prod_{a \in \lambda} \frac{n + c(a)}{h(a)}.$$

Here, the latter expression gives us the **hook-content formula** for counting semi-standard Young tableaux, where $h(a)$ is the usual Young diagram hook length, and $c(a)$ is the **content** of the box a (defined as $j - i$ for a box in the i th row and j th column).

But we've moved away from our original discussion, and we'll now finish by connecting semi-standard Young tableaux back to **reverse plane partitions**, which are those where the entries are **increasing** in rows and columns rather than decreasing. It may seem curious that (when we restrict the set of numbers for our entries) we have an exact product formula for counting **any** shape for Young tableaux, but only really for **rectangular** plane partitions. Here's why that is:

Lemma 304

Reverse plane partitions of rectangular shape $n \times m$, filled with numbers in the set $\{0, 1, \dots, k\}$, are in bijection with semi-standard Young tableaux of shape $n \times m$ filled with $1, 2, \dots, k + n$.

This bijection is easy to describe – we start with a reverse plane partition and add i to all entries in the i th row. However, such a bijection doesn't work for any other shapes! The key fact is that rectangles only have one outer corner, and it's enough to force that the bottom right corner is at most k to force all numbers to be between 0 and k . Unfortunately, with non-rectangles, the multiple corners mess things up, and that means we don't actually have nice explicit formulas for other shapes of reverse plane partitions.

That's all we'll discuss in this course, but there are many books we can read and many other classes in which we can learn more algebraic combinatorics!