18.212 — Algebraic Combinatorics

Class by Alex Postnikov

Notes by Sanjana Das

Spring 2022

Notes for the MIT class **18.212** (Algebraic Combinatorics), taught by Alex Postnikov. All errors are my responsibility.

Contents

§1 Catalan Numbers

Definition 1.1. The nth Catalan number, denoted C_n, is the number of sequences $(\varepsilon_1, \dots, \varepsilon_{2n})$ with the following properties:

- $\varepsilon_i \in \{1, -1\}$ for all i,
- $\varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_{2n} = 0$, and
- $\varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_i \ge 0$ for all i.

Graphically, these sequences can be represented as *Dyck paths* — start at the origin. Every time we have $\varepsilon_i = +1$, go one step up; every time we have $\varepsilon_i = -1$, go one step down. (Go one step right as well.) Then we should end up at $(2n, 0)$, and the third condition means that the path stays weakly above the x-axis.

n	C_n	Sequences
0	1	
1	1	$+-$
2	2	$++--$, $+-+-$
3	5	$+++---$, $++-+--$, $++--+-$, $+-++--$, $+-+-+-$

As we'll see later, the Catalan numbers count many other objects as well: for example, the number of triangulations of a (convex) n-gon, or the number of valid sequences of parentheses. We'll start out by computing what these Catalan numbers are.

Theorem 1.2

The nth Catalan number is given by

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

There are many proofs; we'll look at proofs by generating functions, the reflection principle, and cyclic shifts.

§1.1 Proof by Generating Functions

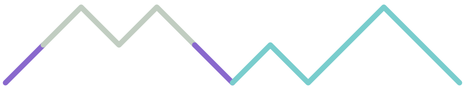
Definition 1.3. Given a combinatorial sequence A_n, its *generating function* is given by

$$A(x) = \sum A_n x^n.$$

We usually treat these as *formal power series* and don't worry about convergence issues — we can do all usual operations (adding, multiplying, and dividing) just with formal power series.

Usually we want to construct a recurrence relation for the sequence — finding a way to decompose the object into smaller objects of the same kind, in order to get an expression for A_n in terms of the previous A_i. This usually translates into an algebraic or differential equation for its generating function, and we can then use analysis or algebra to solve the equation and get a formula for the numbers A_n.

For the Catalan numbers, we want to break a Dyck path of size n into smaller Dyck paths. Consider the first point after the origin where the path touches the x-axis. Let this point be $(2k, 0)$, where $1 \le k \le n$. Then the first segment of the path, from $x = 0$ to $2k$, lies entirely above the x-axis – there's an up-step at the beginning and a down-step at the end, and if you remove these two steps then you get another Dyck path (shifted up by 1). Meanwhile, there's also a path from $(2k, 0)$ to $(2n, 0)$, which is also a Dyck path.



Then the first smaller path has $2(k-1)$ steps, and the second has $2(n-k)$ steps. This decomposition is unique, so there's a correspondence between Dyck paths of $2n$ steps, and these pairs of paths. This means

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

This is the recurrence relation (for all $n \geq 1$), and our initial condition is $C_0 = 1$.

The next step is to convert this into an equation for the generating function. Define

$$C(x) = \sum_{n=0}^{\infty} C_n x^n.$$

We have

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

Now sum over all n, so

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

The left-hand side is $C(x)-1$. On the right-hand side, $k-1$ and $n-k$ can both be any nonnegative number. So then

$$C(x) - 1 = x \cdot C(x) \cdot C(x).$$

So now we have the algebraic equation

$$xC(x)^2 - C(x) + 1 = 0.$$

This is a quadratic equation, so we can solve it and get

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

We almost have the answer, but here we have a $\pm$, which gives us two solutions; so we need to figure out which sign to use. To do so, we can look at $C(0)$ — we know $C(0) = 1$. As $x \to 0$, if we pick $+$ then we get ∞, while if we pick $-$ then we can check that the limit is 1. So then the correct choice is $-$, and

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

Finally, we want to use this to get the expression for C_n. We can use the Binomial Theorem:

Theorem 1.4 (Binomial Theorem)

For all $n \in \mathbb{C}$, we have

$$(1+z)^n = \binom{n}{0}z^0 + \binom{n}{1}z^1 + \binom{n}{2}z^2 + \binom{n}{3}z^3 + \cdots,$$

where we define the *generalized binomial coefficients*

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

We must have $k \in \mathbb{Z}_{\geq 0}$, but n can be any complex number.

If n is not a positive integer, then the series is infinite (when n is a positive integer, everything becomes 0 after a certain point). The proof is basically Taylor expansion.

Now we can apply this to our expression $\sqrt{1-4x}$: we have

$$(1-4x)^{1/2} = \sum_{k\geq 0} \binom{1/2}{k} (-4x)^k.$$

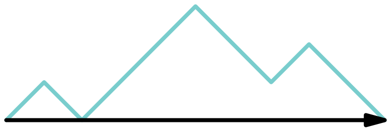
Using $[x^n]$ to denote the coefficient of x^n, we have

$$C_n = [x^n]\left(\frac{1-\sqrt{1-4x}}{2x}\right) = -\frac{1}{2}[x^{n+1}]\left(\sqrt{1-4x}\right) = -\frac{1}{2}\cdot\binom{1/2}{n+1}(-4)^{n+1} = \frac{1}{n+1}\binom{2n}{n}$$

(using the formula for generalized binomial coefficients to compute the final step).

§1.2 Proof by the Reflection Method

We use complementary counting.

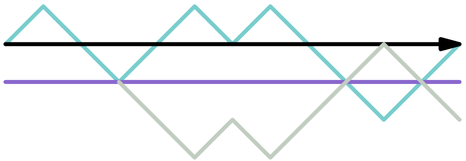


First, the number of arbitrary *lattice paths* from $(0,0)$ to $(2n,0)$ with n up-steps and n down-steps – ignoring the condition that the path must stay above the x-axis – is $\binom{2n}{n}$, since there's n $+1$'s and n -1's. (These can be thought of as paths on an $n \times n$ grid.)

Now we calculate the number of bad paths – the paths which go below the x-axis. Then

$$C_n = \binom{2n}{n} - \#\{\text{paths that go below the } x\text{-axis}\}.$$

A path is bad if and only if it intersects $y = -1$.



Now we transform such a bad path: take the leftmost point X where the path intersects $y = -1$. From a path P, obtain P' by reflecting its portion between X and the endpoint $(2n,0)$ over the line $y=-1$.

Then P' is a lattice path from $(0,0)$ to $(2n,-2)$.

Lemma 1.5

This map $R : P \to P'$ is a bijection between bad paths from $(0,0)$ to $(2n,0)$, and all lattice paths from $(0,0)$ to $(2n,-2)$.

Proof. R has an inverse – given any lattice path from $(0,0)$ to $(2n,-2)$, this path must intersect the line $y=-1$. Then take its first intersection and reflect the remainder of the path over the line $y=-1$. This gives us an inverse R^{-1}. □

But the number of lattice paths from $(0,0)$ to $(2n,-2)$ is $\binom{2n}{n-1}$, since there are $n-1$ up-steps and $n+1$ down-steps. So then this is the number of bad paths as well, and

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

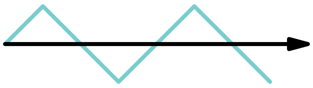
> **Remark 1.6.** This proof is useful for some other problems as well, that we'll see in later problem sets.

§1.3 Proof by Cyclic Shifts

We can actually find a combinatorial proof that doesn't use any subtraction. We want to show that

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{1}{n+1}\cdot\frac{(2n)!}{n!\,n!} = \frac{1}{2n+1}\cdot\frac{(2n+1)!}{n!\,(n+1)!} = \frac{1}{2n+1}\binom{2n+1}{n}.$$

First, $\binom{2n+1}{n}$ is the number of all sequences $(\varepsilon_1, \dots, \varepsilon_{2n+1})$ with n $+1$'s and $n+1$ -1's. Equivalently, in terms of lattice paths, this is the number of lattice paths from the origin to $(2n+1,-1)$, with no other conditions.



We want to show that among all these paths, there are C_n that stay above the x-axis everywhere until the last step – meaning that you take a Dyck path to $(2n,0)$ and add an extra down-step to get to $(2n+1,-1)$. So we want to break paths into groups of $\frac{1}{2n+1}$, where each group has exactly one such path. The way we'll do this is by taking *cyclic shifts*: for example, given our above sequence $(+--++--)$, we'd have the following cyclic shifts (by moving the first entry to the end):

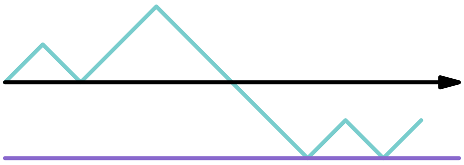
$$(+--++--)$$
$$(--++--+)$$
$$(-++--+-)$$
$$(++--+--)$$
$$(+--+--+)$$
$$(--+--++)$$
$$(-+--++-)$$

Including the original, we get $2n+1$ cyclic shifts.

Lemma 1.7

Exactly one of these $2n+1$ cyclic shifts corresponds to an extended Dyck path.

Proof. Take an arbitrary path from $(0,0)$ to $(2n+1,-1)$.



Taking a cyclic shift means we cut the path and move the first part to the end. We claim that we have to cut at the *leftmost minimal point*. Suppose we cut at a point with $y = d$. Then the second piece starts at $y = d$ and ends up at $y = -1$, so when we move it to the front it starts at $y = 0$ and ends at $y = -d - 1$. So when we move the first part to the end, a point originally at y is moved to $y - d - 1$. If we cut at the first minimum, then $y \geq d + 1$ for all such points, so they all end up weakly above the x-axis; otherwise any $y \leq d$ in the first part corresponds to going below the x-axis. □

Lemma 1.8

All $2n + 1$ shifts are different from each other.

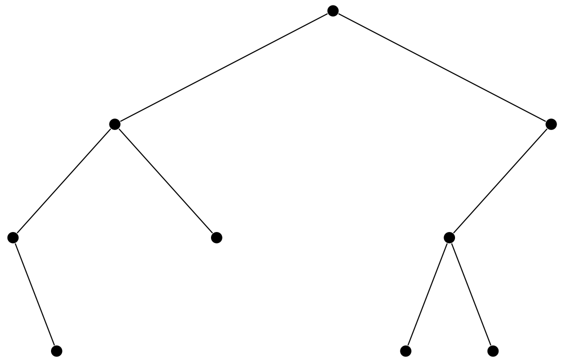
Proof. If two are shifts of each other, then the sequence must be periodic, with period $t \mid 2n + 1$. Then the sum of all entries in the sequence is also divisible by t. But -1 doesn't have any nontrivial divisors, contradiction.

Alternatively, the leftmost minimum also shifts when you take cyclic shifts. □

§1.4 Other Combinatorial Interpretations

The Catalan numbers count a lot of things:

1. The number of Dyck paths of length $2n$.
2. The number of triangulations of a $(n + 2)$-gon (where rotations and reflections are distinct).
3. The number of parenthesizations of $n + 1$ letters (for example, $a((bc)(de))$ is a parenthesization).
4. The number of *plane binary trees* with n vertices – we have a root, and every vertex can have at most one left child, and at most one right child.



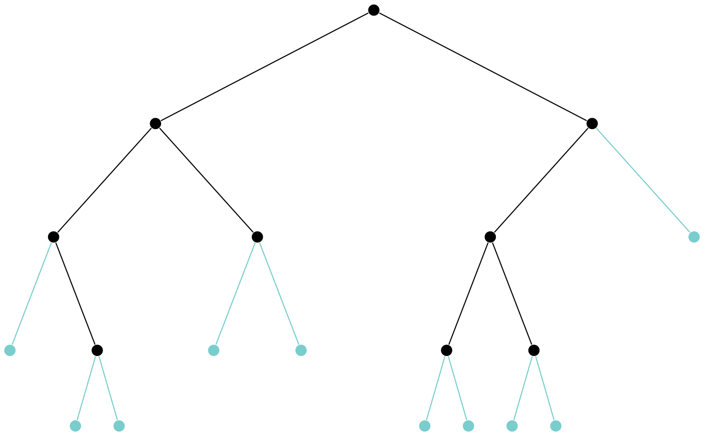
5. The number of *complete* plane binary trees with $n + 1$ *leaves* – a complete binary tree is defined in the same way, but every vertex is either a leaf, or has exactly one left child and one right child.

There are many other objects as well.

Example 1.9

There exists a bijection between plane binary trees with n vertices, and complete plane binary trees with $n + 1$ leaves.

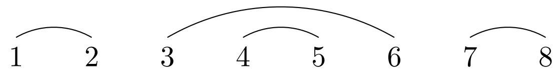
Proof. For every vertex in our tree missing a child, add that child.



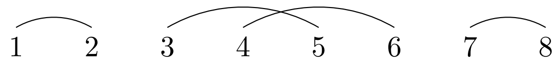
The leaves are exactly the vertices we added, so we can reverse the map by deleting all leaves. □

A few more objects the Catalan numbers count:

6. The number of *plane trees* on $n+1$ vertices: we have a root, and a vertex can have any number of children. By *plane* we mean that we fix a particular drawing of the tree on the plane – the order of the children matter. (But unlike a binary tree, a single child is not designated as the left or right child.)
7. The number of *non-crossing matchings* with $2n$ vertices: we have $2n$ vertices labelled 1 to $2n$ on a line, from left to right, and we break them into pairs by drawing arcs on top of the line. To be non-crossing, none of the arcs should intersect (so we don't have matchings ab and cd with $a < c < b < d$).



8. The number of *non-nesting matchings* with $2n$ vertices: draw the same arc diagrams, but now the arcs are allowed to cross, but not nest (we can't have one arc completely contained in another).



We can construct bijections between these objects.

Example 1.10

There is a bijection between Dyck paths of length $2n$ and non-crossing matchings of $2n$ elements.

Proof. If a vertex is the left element of a pair, then go up; if it's the right element, go down. For example, if we have the pairs 18, 25, 34, and 67, then our path would be $+++--+--$. The inverse is given by labelling the edges of the Dyck path with 1 to $2n$, and matching every left edge to the first right edge on the same level. □

We can similarly construct a bijection between Dyck paths and non-nesting matchings: again have the left number of each pair correspond to a $+$, and right labels to a $-$. But this time, given a Dyck path, connect right labels to the *first* available left label.

Remark 1.11. Here we can see some sort of duality between non-crossing and non-nesting matchings. This relationship shows up in other places as well.

Example 1.12

There is a bijection between triangulations on $n+2$ vertices, and complete plane binary trees with $n+1$ leaves.

Proof. Label the edge between vertex 1 and $n+2$ as special. Place a vertex in every triangle, and a vertex outside every edge of the $(n+2)$-gon except for the special one.

Now if there is an edge between two vertices, connect them. The root is the vertex next to the special edge, and the leaves are the vertices outside the polygon.

To see that this is a bijection, add an extra green vertex and edge corresponding to the special edge. Then the tree and triangulation are dual graphs, which means we can go back from a tree to the triangulation. □

Example 1.13

There is a bijection between plane trees with $n+1$ vertices and Dyck paths with $2n$ steps.

Proof. Given a plane tree, perform a *depth-first search*: go down to the leftmost child until we get stuck, then go back up a step and continue.

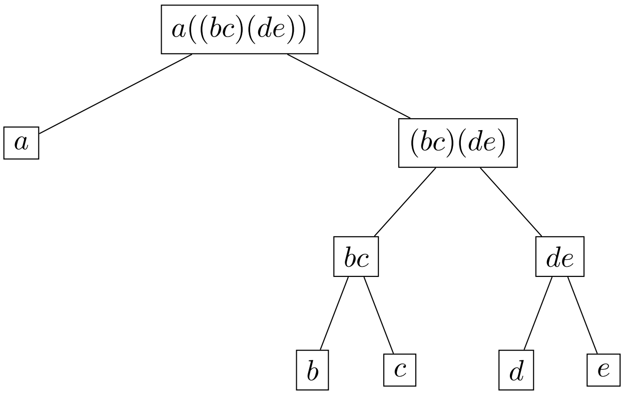
Then turn the up and down steps taken on the tree into up and down steps on a path, and reflect the path over the x-axis. This gives us a Dyck path.

For the inverse, we can match edges the same way as we did earlier (matching a $+$ edge to the first $-$ at the same level) and glue matching edges together – imagine making the underside of the path “sticky” and collapsing it. This recovers the tree. □

Example 1.14

There is a bijection between parenthesizations of $n+1$ letters and complete binary trees with $n+1$ leaves.

Proof. Have each letter be a leaf. Then every time we match two vertices by a pair of parentheses, we make them the left and right child of a new vertex – which we now use to represent their parenthesized group.



For example, the above tree corresponds to $a((bc)(de))$ when $n=4$. □

Using these simple bijections, we can end up breaking all our combinatorial objects into two classes: Dyck paths are easily related to non-crossing and non-nesting matchings and plane trees, while triangulations are related to binary trees and parenthesizations. To go from a triangulation to a Dyck path is slightly more complicated, but a bijection does exist.

§1.5 Pattern Avoidance

Another situation in which the Catalan numbers appear is pattern avoidance.

Consider a permutation of 1 through n, for example $\pi = 63157284$. Consider its subsequences, for example 174, and look at their relative order – here 174 is in the same relative order as 132. We then say that the permutation π *contains* the pattern 132.

Definition 1.15. A permutation π *avoids* a pattern σ (so is σ-avoiding) if it doesn't contain σ.

Theorem 1.16

Fix any pattern σ of size 3. Then the number of σ-avoiding permutations π with n letters is C_n.

We won't prove this; it'll be on the first problem set.

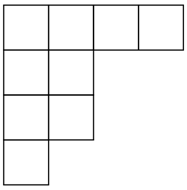
§2 Young Tableaux

§2.1 Partitions and Young diagrams

Definition 2.1. $\lambda = (\lambda_1, \dots, \lambda_k)$ is a *partition* of n if the λ_i are integers, $\lambda_1 \geq \cdots \geq \lambda_i > 0$, and $\lambda_1 + \cdots + \lambda_k = n$.

We use $\lambda \vdash n$ to denote that λ is a partition of n. Here k is the number of parts of the partition.

With each partition, we can associate with it a *Young diagram*, where the number of boxes in row i is λ_i. For example, $\lambda = (4, 2, 2, 1) \vdash 9$ corresponds to the following Young diagram:



Definition 2.2. A *Standard Young Tableau* (SYT) of shape $\lambda \vdash n$ is a filling of the boxes of λ with 1, 2, . . . , n, without repetition, such that the entries increase in both rows and columns.

For example, one SYT of shape λ is the following:

1	2	3	7
4	5		
6	9		
8			

We can think of a SYT as a way to grow our Young diagram one square at a time.

For a partition $\lambda \vdash n$, we use f_λ to denote the number of SYT of shape λ.

The f_λ are important in a lot of places – they are the dimensions of irreducible representations of S_n.

Example 2.3

A few examples of f_λ:

- $f_\varnothing = 1$, where $\varnothing$ is the empty partition of 0, since there's nothing to fill.
- $f_{(4)} = 1$ since there's only one way to fill the row.
- $f_{(2,1)} = 2$, since 1 must go in the top-left corner, and then we can make either choice for 2 and 3.

Proposition 2.4

We have $f_{(n,n)} = C_n$ for all n.

Proof. There is a bijection between SYT of shape (n, n) and Dyck paths of length $2n$: if i is in the first row, then take the ith step to be up, while if it's in the second row, then take the ith step to be down. For example, the SYT

1	2	3	6	8	10
4	5	7	9	11	12

corresponds to the path

$$+ + + - - + - + - - + - .$$

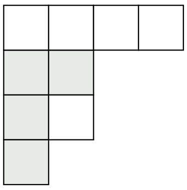
There are n up-steps and n down-steps. Every time you add a number, there's at least as many entries in the first row as the second, which exactly corresponds to the path staying above the x-axis. □

We can think of f_λ sort of as generalized Catalan numbers. For example, (n, n, n) is the number of "3-dimensional Dyck paths" from the origin to (n, n, n) – going in the x, y, and z directions instead of up and down – such that $x \geq y \geq z$ at all times.

§2.2 The Hook Length Formula

It turns out there is a formula for f_λ.

Definition 2.5. Given a box a of a Young diagram λ, the *hook* of a is all boxes directly to the right of or below a, as well as a itself. The *hook length* of a, denoted $h(a)$, is the number of boxes in its hook.



For example, the above hook length is 4.

Definition 2.6. For a Young diagram λ, define

$$H(\lambda) = \prod_{a \in \lambda} h(a)$$

to be the product of all hook lengths.

Then we have the following formula:

Theorem 2.7 (Hook Length Formula)

For any $\lambda \vdash n$, we have

$$f_\lambda = \frac{n!}{H(\lambda)}.$$

Example 2.8

Find $f_{(4,2,2,1)}$.

Solution. We can calculate the hook lengths of all squares:

7	5	2	1
4	2		
3	1		
1			

So the answer is

$$f_{(4,2,2,1)} = \frac{9!}{7\cdot 5\cdot 1\cdot 1\cdot 4\cdot 2\cdot 3\cdot 1\cdot 1} = \boxed{216}.$$ □

Example 2.9

Find $f_{(n,n)}$.

Solution. The hook lengths in the top row are $n+1$, n, ..., 2, while the hook lengths in the bottom row are n, $n-1$, ..., 1. So the formula gives

$$f_{(n,n)} = \frac{(2n)!}{n!\,(n+1)!} = \frac{1}{n+1}\binom{2n}{n},$$

which is the nth Catalan number. □

The Hook Length Formula is not easy to prove; we will prove it by a *probabilistic hook walk.*

§2.2.1 Proof by Probabilistic Hook Walk

We will give a proof of HLF found by three authors – Greene, Nijenhuis, and Wilf – in 1979.

Lemma 2.10

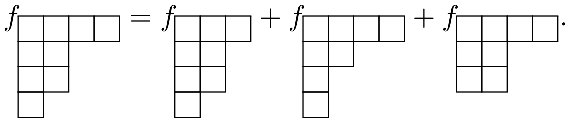
We have a simple recurrence relation

$$f_\lambda = \sum_c f_{\lambda - c},$$

where the sum is over all corners c of λ, and $\lambda - c$ is λ with c removed. The initial condition is $f_\varnothing = 1$.

The corner boxes are the boxes with hook length 1 — these are exactly the boxes which we can remove from the Young diagram so that it remains a Young diagram. So this lemma is true simply by casework on where we place n (which must be a corner).

For example, we have



This gives us a (very inefficient) way to calculate f_λ. Our goal is to prove that the same recurrence relation holds for the right-hand side of HLF — then by induction, the two sequences must be equal.

First, for the empty shape we have $0!\,/H(\varnothing) = 1$, so the initial condition is satisfied.

Proposition 2.11

The right-hand side of HLF satisfies the same recurrence as in Lemma 2.10, meaning

$$\frac{n!}{H(\lambda)} = \sum_c \frac{(n-1)!}{H(\lambda - c)},$$

where the sum is over all corners c of λ.

In order to prove this, we can rewrite this equation as

$$1 = \sum_c \frac{1}{n} \cdot \frac{H(\lambda)}{H(\lambda - c)}.$$

Note that many of the hook lengths in $H(\lambda)$ and $H(\lambda - c)$ are the same: only a small number of terms change by 1, and the rest cancel.

If we have several numbers which add up to 1, this may remind us of probability. So the idea is to construct a random process that outputs a corner of λ, such that the probability of outputting c is exactly the expression in the sum. The process we'll use is the *Hook Walk*:

> **Algorithm 2.12** (Hook Walk) — First, randomly pick any box $a_1 \in \lambda$, with uniform probability. Then if we are currently at box a_i, jump to any box in the hook of a_i other than a_i itself, with uniform probability. Repeat until we arrive at a corner c; then stop and output c.

Let $\mathbb{P}(c)$ be the probability that the random walk ends at c.

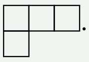
Proposition 2.13

The probability that we arrive at a corner c is exactly

$$\mathbb{P}(c) = \frac{1}{n} \cdot \frac{H(\lambda)}{H(\lambda - c)}.$$

Once we show this, we immediately get the identity we want, because the sum of all probabilities is 1. So it suffices to prove this proposition.

Example 2.14

Consider $\lambda =$ . Find the probability of landing in each corner.

Solution. Label the corner boxes c_1 and c_2.

If we start at the second or third boxes in the first row, then we always end up at c_1. If we pick the first box in the first row, then to end up at c_1 we must jump right. If we pick c_2, then we can't reach c_1. So then

$$\mathbb{P}(c_1) = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} \cdot \frac{2}{3} + 0 = \frac{2}{3}.$$

Similarly we have

$$\mathbb{P}(c_2) = \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{3} = \frac{1}{3}.$$

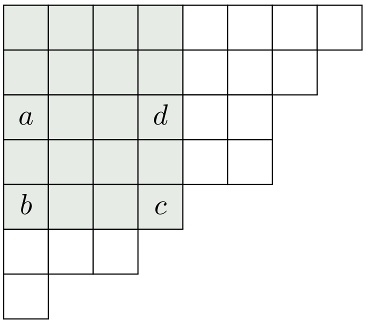
We can calculate that these are the correct values of $H(\lambda)/nH(\lambda - c)$, which verifies the claim in this case. □

In general, we have

$$\mathbb{P}(c) = \frac{1}{n} \sum_{a_1, \ldots, a_k = c} \frac{1}{h(a_1) - 1} \cdot \frac{1}{h(a_2) - 1} \cdots \frac{1}{h(a_{k-1}) - 1}.$$

(We are allowing this product to be empty, with $k = 1$, corresponding to starting on the corner.) This is because each jump has probability $1(h(a_i) - 1)$ of happening.

If we end up on c, then we must remain in the rectangle with bottom-left corner c.



For any square a inside this rectangle, let b and d be the other corners of the rectangle with top-left corner a and bottom-right corner c. Then we can check that

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

Since $h(c) = 1$, we get

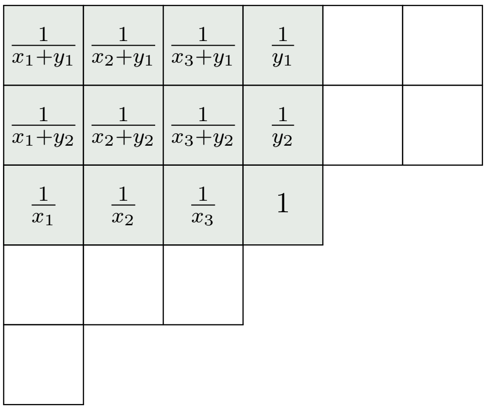
$$h(a) - 1 = (h(b) - 1) + (h(d) - 1).$$

This means if we know the hook lengths of all boxes in the same row and column as our corner, then we can express the hook length for any other box.

Now let $x_1, \ldots, x_k$ and $y_1, \ldots, y_\ell$ denote $h(a) - 1$ for the row and column of c, respectively. Let the *weight* of a box a be

$$w(a) = \frac{1}{h(a) - 1},$$

where $w(c) = 1$. Then we have a $(k + 1) \times (\ell + 1)$ rectangle. The weights of boxes on the row and column of c are $\frac{1}{x_i}$ and $\frac{1}{y_j}$, and the weights of other boxes in the rectangle are $\frac{1}{x_i + y_j}$.



For a walk P, denote the weight of P as the product of the weights of its boxes:

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

Then $w(P)$ is the probability of our path being P, so we want to sum $w(P)$ over all paths ending at c.

Lemma 2.15

If we sum over all possible *lattice paths* P in the blue $(k+1)\times(\ell+1)$ rectangle,

$$\sum_P w(P) = \frac{1}{x_1x_2\cdots x_ky_1y_2\cdots y_\ell}.$$

A lattice path is a path where we start in the top-left corner, and all squares are consecutive – we move one square right or down in each step. This is a subset of all the hook walks.

$\frac{1}{x_1+y_1}$	$\frac{1}{y_1}$
$\frac{1}{x_1}$	1

For example, if $k=\ell=1$ then we have

$$\frac{1}{(x_1+y_1)\cdot y_1\cdot 1}+\frac{1}{(x_1+y_1)\cdot x_1\cdot 1}=\frac{1}{x_1y_1}.$$

Proof. Induct on $k+\ell$, where the base case $k=\ell=0$ is trivially true. For the inductive step, split the sum into paths with the first step right, and paths with the first step down.

Either way, the corner contributes $\frac{1}{x_1+y_1}$. In the first case, the rest of the path is a lattice path of the $k\times(\ell+1)$ rectangle where we delete the first column, which has sum $\frac{1}{x_2\cdots x_ky_1\cdots y_\ell}$. Similarly, the second case has sum $\frac{1}{x_1\cdots x_ky_2\cdots y_\ell}$. So we have

$$\sum_P w(P) = \frac{1}{x_1+y_1}\cdot\left(\frac{1}{x_2\cdots x_ky_1\cdots y_\ell}+\frac{1}{x_1\cdots x_ky_2\cdots y_\ell}\right)=\frac{1}{x_1\cdots x_ky_1\cdots y_\ell},$$

as desired. □

Now we can use this to get a sum over *all* hook walks.

Lemma 2.16

If we sum over all hook walks in the $(k+1)\times(\ell+1)$ rectangle,

$$\sum_P w(P) = \prod_{i=1}^k\left(1+\frac{1}{x_i}\right)\prod_{j=1}^\ell\left(1+\frac{1}{y_j}\right).$$

For example, if $k=\ell=1$, then our sum is

$$1+\frac{1}{x_1}+\frac{1}{y_1}+\frac{1}{(x_1+y_1)x_1}+\frac{1}{(x_1+y_1)y_1}=\left(1+\frac{1}{x_1}\right)\left(1+\frac{1}{y_1}\right).$$

Proof. Use the first lemma. For each hook walk, there is a set of rows and columns which it passes through. Then delete all other columns – take the subrectangle formed by the rows and columns that our walk passes through.

The x_i and y_j do not change, and our hook walk is a lattice path on the subrectangle. So then we're summing $\frac{1}{\prod x_i\prod y_i}$ (where the products are over the rows and columns of the subrectangle) over all choices of subrectangles. This is exactly what we get when we expand the product – taking 1 corresponds to not including row i, while taking $\frac{1}{x_i}$ corresponds to including it. □

But this lemma is exactly what we need to finish. We have

$$\mathbb{P}(c) = \frac{1}{n}\sum_P w(P) = \frac{1}{n}\prod_{i=1}^{k}\left(1+\frac{1}{x_i}\right)\prod_{j=1}^{\ell}\left(1+\frac{1}{y_j}\right).$$

But the x_i and y_j are the hook lengths minus one. If we define the *cohook* C of c to be the squares to the left of or above c, then this sum is

$$\frac{1}{n}\prod_{a\in C}\frac{h(a)}{h(a)-1}.$$

But this product is exactly

$$\frac{H(\lambda)}{H(\lambda-c)},$$

since the boxes in the cohook are exactly the ones whose hook lengths decrease by 1 when we delete c. So this proves $\mathbb{P}(c)$ is the expression we wanted, and we're done.

§3 Set Partitions

Notation 3.1. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

Definition 3.2. A *set partition* of $[n]$ is a way to subdivide $[n]$ as a disjoint union of nonempty blocks.

Example 3.3

One set partition of $[9]$ is

$$\pi = (1, 3, 4, 7 \mid 2, 9 \mid 5 \mid 6, 8).$$

We don't care about the order of blocks, or the order of elements within blocks: this is the same partition as $(8, 6 \mid 5 \mid 1, 4, 3, 7 \mid 2, 9)$, for example.

We can think of set partitions as a labelled version of integer partitions. If we have 9 unlabelled dots, then we have the integer partition $(4, 2, 2, 1)$ corresponding to the number of dots in each group. On the other hand, if our dots are labelled 1 through n, then we get a set partition.

There are a lot of objects in combinatorics with a labelled and unlabelled version.

Definition 3.4. If we have a sequence a_0, a_1, a_2, $\ldots$, then its *ordinary generating function* is the sum

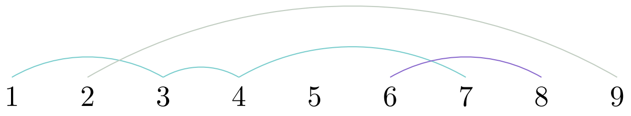
$$\sum a_n x^n,$$

and its *exponential generating function* is the sum

$$\sum \frac{a_n x^n}{n!}.$$

These are the most common types of generating functions, and which one we use depends on the combinatorial object. Generally, we use ordinary generating functions for unlabelled objects, and exponential generating functions for labelled ones. (This is because of how multiplication works, for example.) For example, there is a nice ordinary generating function for integer partitions, and a nice exponential one for set partitions (we will see both later).

We can also represent set partitions in arc notation: write 1 through n in a line, and write each block in increasing order and draw a sequence of arcs for each block. For example, our above set partition of $[9]$ has the following arc diagram:



So in an arc diagram, we have a bunch of chains; every vertex has at most one arc going left, and at most one arc going right.

§3.1 Rook Placements

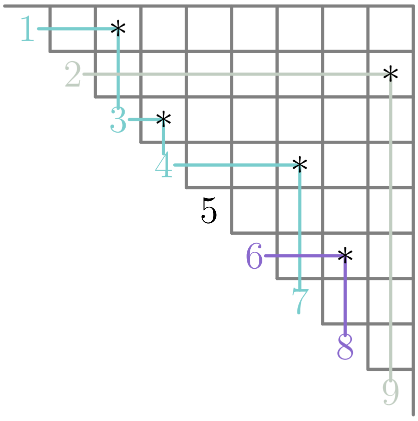
Definition 3.5. Suppose we have a triangular chessboard, with $n-1$ boxes in the first row, $n-2$ in the second, and so on. A *rook placement* is a placement of some (possibly zero) rooks on the chessboard, so that no two rooks attack each other.

We can label the corners of the chessboard with 1 through n.

Proposition 3.6

There is a bijection between rook placements and set partitions of $[n]$.

Proof. For each rook, place the number to its left and the number below it in the same block.



Then the "hooks" of the rooks correspond exactly to arcs. □

In this bijection, the number of blocks is n minus the number of rooks.

§3.2 Bell and Stirling Numbers

Definition 3.7. The Bell number $B(n)$ is the number of set partitions of $[n]$.

Definition 3.8. The Stirling number of the second kind $S(n,k)$ is the number of set partitions of $[n]$ with exactly k blocks.

Equivalently, $S(n,k)$ is the number of rook placements with $n-k$ rooks.

We can calculate:

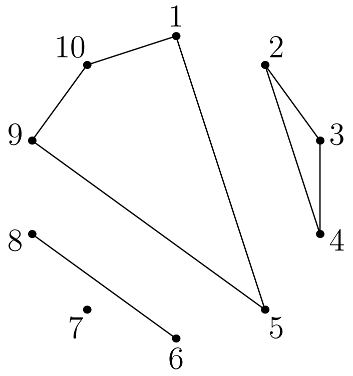
n	$B(n)$	Set Partitions
0	1	
1	1	(1)
2	2	(12), (1 | 2)
3	5	(123), (1 | 23), (2 | 13), (3 | 12), (1 | 2 | 3)

The next two numbers are 15 and 52. So these start out the same as the Catalan numbers, but are not the same sequence – notice though that the Bell numbers are *at least* the Catalan numbers.

§3.3 Non-crossing and Non-nesting Set Partitions

Definition 3.9. A *non-crossing set partition* is a set partition such that if we write 1 through n on a circle, the polygons corresponding to each set do not intersect.

Polygons are allowed to consist of one or two vertices. For example, for $n = 10$:



We can also interpret non-crossing set partitions via arc diagrams: imagine cutting the circle between 1 and n. Then the chains correspond to polygons (where we take the perimeter of the polygon and remove one edge). Then π is non-crossing iff its arcs don't cross.

Definition 3.10. A *non-nesting set partition* is a set partition π whose arcs don't nest.

So we can't have one arc entirely below another arc, but having a singleton below an arc is okay.

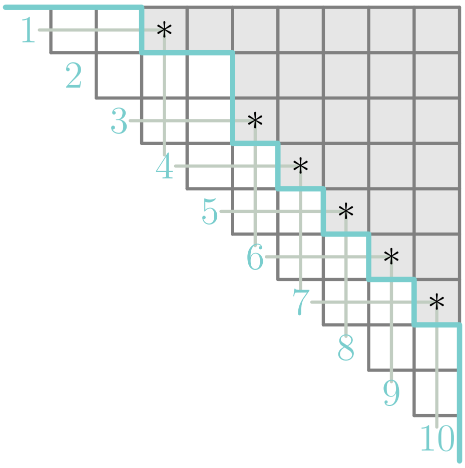
n	Number of non-crossing	Number of non-nesting
0	1	1
1	1	1
2	2	2
3	5	5
4	14	14

Theorem 3.11

The number of non-crossing set partitions of $[n]$ and the number of non-nesting set partitions of $[n]$ are both the nth Catalan number C_n.

Proof. We'll show a bijection for non-nesting partitions, using rook placements.

For example: take $\pi = (1, 4, 7, 10 \mid 2 \mid 3, 6, 9 \mid 5, 8)$. This corresponds to a rook placement, where the rooks represent arcs: the non-nesting condition means that no rook is southwest of another rook.



Now draw a sun in the bottom-left, and have each rook cast a shadow – the rectangle whose bottom-left corner is the rook. Then there is a path separating the illuminated region from the shadowed region. If we append one step to the left at the beginning and one step down at the end, then we get a rotated Dyck path with $2n$ steps. □

The number of blocks in π is n minus the number of rooks. Meanwhile, we can think of a Dyck path as a mountain range, with peaks and valleys. There is one rook in each valley, so the number of blocks is n minus the number of valleys, or $n+1$ minus the number of peaks.

Stirling numbers refine Bell numbers, so we can refine Catalan numbers similarly:

Definition 3.12. The Narayana number $\mathrm{N}(n,k)$ is the number of Dyck paths with $2n$ steps and k peaks.

More formally, a peak is a point where the path is $+-$.

Theorem 3.13

The number of non-crossing set partitions of $[n]$ with k blocks, and the number of non-nesting set partitions of $[n]$ with k blocks, are both $\mathrm{N}(n,k)$.

Example 3.14

Calculate all three numbers for $(n,k)=(4,2)$.

Solution. For $\mathrm{N}(n,k)$, we want to find Dyck paths with 8 steps and 2 peaks. These are:

$$\begin{gathered}+-+++---\\++--++--\\++-++---\\+++---+-\\+++--+--\\+++-+---\end{gathered}$$

So we have $\mathrm{N}(4,2)=6$.

For non-crossing set partitions with two blocks, we want two polygons on four vertices, which don't cross. We can take a triangle and singleton (4 ways), or two opposite chords (2 ways), giving 6 ways.

For non-nesting set partitions, the possibilities are:

$$\begin{gathered}(123\mid 4),\\(1\mid 234),\\(124\mid 3),\\(134\mid 2),\\(12\mid 34),\\(13\mid 24).\end{gathered}$$

This gives us 6 options as well. □

Remark 3.15. For non-nesting partitions, we saw a bijection last lecture where the number of blocks is n minus the number of rooks. This gives us $\mathrm{N}(n, n+1-k)$. But the Narayana numbers have the symmetry

$$\mathrm{N}(n,k) = \mathrm{N}(n, n-k+1).$$

This isn't obvious, and this symmetry isn't true for Stirling numbers.

§3.4 The Stirling Triangle

We would like a more efficient way to calculate these numbers. We can use the *Stirling triangle of the 2nd kind* (which is the same configuration as Pascal's Triangle):

$$\begin{array}{ccccc}
 & & S(0,0) & & \\
 & S(1,0) & & S(1,1) & \\
S(2,0) & & S(2,1) & & S(2,2)
\end{array}$$

We can calculate the first few terms as:

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 0 & & 1 & & & & \\
 & & & 0 & & 1 & & 1 & & & \\
 & & 0 & & 1 & & 3 & & 1 & & \\
 & 0 & & 1 & & 7 & & 6 & & 1 & \\
0 & & 1 & & 15 & & 25 & & 10 & & 1
\end{array}$$

Proposition 3.16

We have the recurrence

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

This is a similar recurrence to in Pascal's Triangle, but we multiply the right entry by a coefficient (corresponding to what diagonal we're on).

Proof. If n is in its own block, then there are $S(n-1,k-1)$ set partitions, since the remaining $n-1$ must form $k-1$ blocks. Meanwhile, if n is not in its own block, then the remaining entries must form k blocks – in $S(n,k)$ ways – and there are k choices for which block to place n in, giving $kS(n,k)$ set partitions. □

§3.5 Generating Functions

We can find an exponential generating function for the Bell numbers

$$\sum_{n\geq 0} \frac{B(n)x^n}{n!}.$$

Theorem 3.17 (Exponential Formula)

Suppose c_n counts some kind of *connected* objects (call these C-objects) on n labelled vertices, and d_n counts the number of objects on n labelled vertices where each connected component is a C-object. Then if

$$c(x) = \sum_{n\geq 1} \frac{c_n x^n}{n!} \text{ and } d(x) = \sum_{n\geq 0} \frac{d_n x^n}{n!},$$

their exponential generating functions are related by

$$d(x) = e^{c(x)}.$$

For example, if c_n counts the number of trees on n labelled vertices, then d_n counts the number of forests.

Proof. By definition, we have

$$d_n = \sum_{\pi=(B_1|\cdots|B_k)} c_{|B_1|}\cdots c_{|B_k|}.$$

Here the sum is over all set partitions π (where order of the blocks doesn't matter). We can rewrite this as

$$\sum_{k\geq 0} \frac{1}{k!} \sum_{n_1+\cdots+n_k=n} \binom{n}{n_1, n_2, \ldots, n_k} c_{n_1}\cdots c_{n_k}.$$

The $k!$ is because we overcount for the reordering of parts. (Even if $n_1 = n_2$, the *blocks* corresponding to those, which we get with the multinomial coefficient, are different.) Here all the n_i are positive.

Now we can write

$$\frac{d_n}{n!}x^n = \sum_{k\geq 0} \frac{1}{k!} \sum_{n_1+\cdots+n_k=n} \frac{c_{n_1}}{n_1!}x^{n_1}\cdots\frac{c_{n_k}}{n_k!}x^{n_k}.$$

Now we can sum over all nonnegative values of n, to get

$$\sum_{n\geq 0} \frac{d_n}{n!}x^n = \sum_{k\geq 0} \frac{1}{k!} \sum_{n_i\geq 1} \left(\frac{c_{n_1}}{n_1!}x^{n_1}\cdots\frac{c_{n_k}}{n_k!}x^{n_k}\right).$$

We can swap the sum and write this as

$$d(x) = \sum_{k\geq 0} \frac{1}{k!} \left(\sum_{n\geq 1} \frac{c_n x^n}{n!}\right)^k = \sum \frac{1}{k!} c(x)^k = e^{c(x)}.$$

□

Corollary 3.18

We have

$$\sum_{n\geq 0} B(n)\frac{x^n}{n!} = e^{e^x-1}.$$

Proof. If the c_n are 1 for each n, then d_n is the Bell number. We then have

$$c(x) = \sum_{n\geq 1} \frac{x^n}{n!} = e^x - 1,$$

which means

$$B(x) = e^{e^x-1}$$

by the Exponential Formula. □

We can actually generalize the exponential formula.

Theorem 3.19

Suppose that $d_{n,k}$ is the number of C-objects on n labelled vertices, with exactly k connected components. Then $d_{n,k}$ has the exponential generating function

$$d(x, y) = \sum_{n\geq 0}\sum_{k\geq 0} d_{n,k}\frac{x^n}{n!}y^k = e^{y\cdot c(x)}.$$

The proof is basically the same – just add appropriate powers of y.

Remark 3.20. Note that the generating function for c starts at $n = 1$, while the sum for d starts at $n = 0$. We assume there is one empty object, and that empty object is not connected.

If we want to define $e^z = 1 + z + \frac{z^2}{2} + \cdots$, and we plug another series without a constant term, then for any power of x there's finitely many terms – so we have a well-defined operation of exponentiation of formal power series. But if there is a constant term, then we can get infinitely many.

So this gets the generating functions for Stirling Numbers as well:

$$S(x, y) = \sum_{n\geq 0}\sum_{k\geq 0} S(n, k)\frac{x^n}{n!}y^k = e^{y(e^x-1)}.$$

Example 3.21

Suppose c_n is 1 if $n = 2$, and 0 otherwise. Then d_n is the number of perfect matchings on n labelled vertices. Find d_n.

Proof. We have $c(x) = \frac{x^2}{2}$, which means

$$d(x) = e^{x^2/2} \implies \frac{d_{2m}}{(2m)!} = [x^{2m}]e^{x^2/2} = \frac{1}{m!}\cdot\frac{1}{2^m}.$$

This means

$$d_{2m} = \frac{(2m)!}{m!\,2^m} = \frac{1\cdot 2\cdot 3\cdots(2m)}{2\cdot 4\cdot 6\cdots(2m)} = 1\cdot 3\cdot 5\cdots(2m-1) = (2m-1)!!.$$

There is a simple combinatorial reason: there's $2m-1$ ways to match the first number 1, then $2m-3$ ways to match the next number, and so on (matching the smallest unmatched number at each step). □

§4 Permutations

Definition 4.1. A *permutation* is a rearrangement $w = w_1 \dots w_n$ of $[n]$.

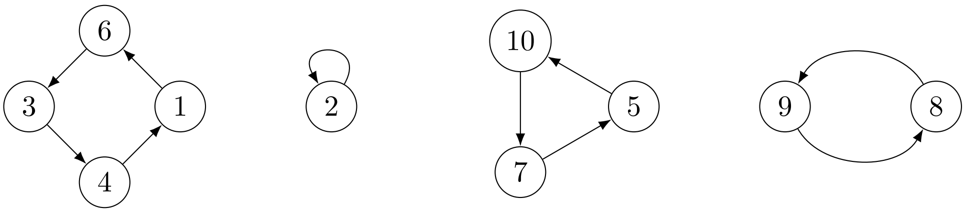
Definition 4.2. The set of all permutations of n letters is the *symmetric group* S_n.

There are several ways to write down a permutation: for example, we can also think of permutations as bijective maps $w : [n] \to [n]$ (where $i \mapsto w_i$). Some common representations include:

1. 1-line notation is the representation of w as a word: for example, $w = (6, 2, 4, 1, 10, 3, 5, 9, 8, 7)$.
2. In 2-line notation, we write 1 through n in order in the first row, followed by the word in the second row. This represents w as a bijection. For example,

$$w = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 2 & 4 & 1 & 10 & 3 & 5 & 9 & 8 & 7 \end{bmatrix}.$$

3. We can represent a permutation as a directed graph, by drawing arrows $i \to w_i$. The graph must always consist of several cycles (since all in-degrees and out-degrees are 1). For example:



4. We can use *cycle notation* by writing w as the product of cycles. For example:

$$w = (1, 6, 3, 4)(2)(5, 10, 7)(8, 9).$$

(We sometimes leave out fixed points in cycle notation.)

§4.1 Stirling Numbers

We can study permutations by their number of cycles.

Notation 4.3. Define $\operatorname{cyc}(w)$ to be the number of cycles in a permutation w, including fixed points.

For example, our above permutation has $\operatorname{cyc}(w) = 4$.

Definition 4.4. The signless Stirling number of the first kind $c(n, k)$ is the number of permutations $w \in S_n$ with exactly k cycles. The signed Stirling number of the first kind is

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

§4.1.1 Stirling Triangle of the First Kind

We saw that the Stirling numbers of the second kind form a triangle. There is a similar triangle for Stirling numbers of the first kind.

Again arrange the (signless) Stirling numbers in a triangle, where the first row is $c(0, 0)$, the second is $c(1, 0)$ and $c(1, 1)$, and so on.

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 0 & & 1 & & & & \\
 & & & 0 & & 1 & & 1 & & & \\
 & & 0 & & 2 & & 3 & & 1 & & \\
 & 0 & & 6 & & 11 & & 6 & & 1 & \\
0 & & 24 & & 50 & & 35 & & 10 & & 1
\end{array}$$

Proposition 4.5

We have the recurrence

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

So every number is the parent on its left, plus $n-1$ times the parent on the right. Here the multiplicity of the right edge depends on the row, while in the triangle for Stirling numbers of the second kind, it depended on the diagonal.

Proof. If n is a fixed point this gives us $c(n-1,k-1)$ ways, since the remainder must have $k-1$ cycles. Otherwise, if n is in a cycle containing other elements, then there are $c(n-1,k)$ ways to form the cycles on the other vertices, and then $n-1$ possible places in which we can add n (right after any of the smaller elements). This gives a total of $c(n-1,k-1)+(n-1)c(n-1,k)$ ways. □

§4.1.2 Stirling Number Relations

It turns out that the two kinds of Stirling numbers are actually related:

Theorem 4.6

We have the sums

$$\sum_{k=0}^{n} s(n,k)x^k = x_{(n)} \text{ and } \sum_{k=0}^{n} S(n,k)x_{(k)} = x^n.$$

Notation 4.7. The nth falling power of x (or falling factorial) is

$$x_{(n)} = x(x-1)(x-2)\cdots(x-n+1).$$

Similarly

$$x^{(n)} = x(x+1)(x+2)\cdots(x+n-1)$$

is the nth rising power (or rising factorial).

Example 4.8

Take $n=3$, and verify both identities.

Proof. For the first, we get

$$-0 + 2\cdot x - 3\cdot x^2 + x^3 = x(x-1)(x-2),$$

which is true. For the second, we get

$$0 + 1\cdot x + 3\cdot x(x-1) + 1\cdot x(x-1)(x-2) = x^3,$$

which is also true. □

One way of thinking about these identities is that there are two bases of the space of polynomials in x: one given by the ordinary powers x^i, and another given by the falling powers $x_{(i)}$. Then the first Stirling numbers give a way to go from the first basis to the second, and the second Stirling numbers give a way to go in the opposite direction.

Corollary 4.9

For any n, we can arrange the Stirling numbers of the first kind in a lower triangular $(n+1)\times(n+1)$ matrix, where $a_{ij} = s(i,j)$.

Then the inverse of this matrix is the matrix with $a_{ij} = S(i,j)$.

There are several ways to prove these formulas. One is to derive them from the exponential formula (since permutations are a collection of cycles). There are also more interesting combinatorial arguments.

Proof of first identity. By flipping the sign of x, the identity is equivalent to

$$\sum_{w\in S_n} x^{\operatorname{cyc}(w)} = \sum_{k=0}^{n} c(n,k)x^k = x^{(n)} = x(x+1)(x+2)\cdots(x+n-1).$$

Now we can induct on n: we have

$$\sum_{k=0}^{n} c(n,k)x^k = \left(\sum_{\ell=0}^{n-1} c(n-1,\ell)x^\ell\right)(x+(n-1))$$

by the recursion we proved earlier, which gives the desired formula. □

Proof of second identity. We want to show

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

This is a polynomial identity, so it suffices to prove it for all positive integers x.

The right-hand side is the number of functions $f : [n] \to [x]$. Meanwile, the left-hand side is

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

But $k!\,S(n,k)$ is the number of *ordered* set partitions of $[n]$ with k blocks. So the left-hand side is the number of ways to choose a subset $\{a_1,\dots,a_k\} \in [x]$ and a *surjective* map $f : [n] \to \{a_1,\dots,a_k\}$. So this counts the number of functions $f : [n] \to [x]$ as well, counting functions by their image. □

§4.2 Statistics on Permutations

Definition 4.10. A *statistic* on permutations is a function $\sigma : S_n \to \mathbb{Z}_{\geq 0}$.

For example, $\operatorname{cyc}(w)$ is a statistic.

Given a statistic, we can write its generating function

$$F_\sigma(x) = \sum_{w\in S_n} x^{\sigma(w)}.$$

Definition 4.11. Two statistics σ and μ are *equidistributed* if $F_\sigma = F_\mu$. We denote this $\sigma \sim \mu$.

Equivalently, two statistics are equidistributed if for all k, the statistics have an equal number of permutations which are mapped to k.

Definition 4.12. A few important statistics on permutations:

1. An *inversion* is a pair of indices (i, j) with $1 \le i < j \le n$ and $w_i > w_j$. The number of inversions is $\operatorname{inv}(w)$.
2. A *descent* is an index $1 \le i \le n-1$ such that $w_i > w_{i+1}$. The number of descents is $\operatorname{des}(w)$, and the sum of indices of the descents is $\operatorname{maj}(w)$ (called the Major Index).
3. A *record* of w is an index i such that $w_i > w_j$ for all $j < i$. The number of records is $\operatorname{rec}(w)$.
4. A *exceedence* of w is an index i such that $w_i > i$. The number of exceedences is $\operatorname{exc}(w)$.

Example 4.13

Calculate these statistics on the permutation 25731684.

Solution. The cycle notation of w is $(125)(3784)(6)$, so $\operatorname{cyc}(w) = 3$. We can count $\operatorname{inv}(w) = 11$. There are descents at indices 3, 4, and 7, so $\operatorname{des}(w) = 3$ and $\operatorname{maj}(w) = 14$. The records are the numbers 2, 5, 7, and 8 (at indices 1, 2, 3, and 7), so $\operatorname{rec}(w) = 4$. Finally, the exceedences are the indices 1, 2, 3, and 7, so $\operatorname{exc}(w) = 4$ as well. (Fixed points are not exceedences, although we can define *weak exceedences* to include fixed points.) □

It happens that many of these statistics are equidistributed.

Theorem 4.14

The following pairs of statistics are equidistributed:

1. $\operatorname{inv} \sim \operatorname{maj}$,
2. $\operatorname{cyc} \sim \operatorname{rec}$,
3. $\operatorname{des} \sim \operatorname{exc}$.

Many statistics belong to one of these three classes. Statistics in the first class are called *Mahonian*, and statistics in the third are called *Eulerian*.

Proof of 2. We'll find a bijection $S_n \to S_n$ sending $w \to \tilde{w}$, such that $\operatorname{cyc}(w) = \operatorname{rec}(\tilde{w})$.

Write w in cycle notation, and arrange the cycles in ascending order of the largest element, with the largest element written first. (Since we can reorder cycles and cyclically shift indices, there is a unique way to do so.) Then erase all the parentheses. For example, our w gives

$$(125)(3784)(6) \to (512)(6)(8437) \to 51268437.$$

Then $\tilde{w}$ has exactly one record per cycle (its first element). But given $\tilde{w}$, we can reconstruct w – the splits between cycles are exactly before each record. So this is a bijection. □

Remark 4.15. A similar bijection can prove other pairs of equidistributed statistics as well.

§5 Posets

Theorem 5.1 (Sperner)
Suppose $S_1, \ldots, S_N$ are distinct subsets of $[n]$, such that for any $i \neq j$, $S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$. Then

$$N \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

This bound is tight, since we can take all subsets of cardinality $\lfloor \frac{n}{2} \rfloor$.

In order to prove this theorem, we will develop the theory of posets.

§5.1 Definitions

Definition 5.2. A *poset* (partially ordered set) is a pair $(P, \leq)$ satisfying the following axioms:

- $a \leq a$;
- If $a \leq b$ and $b \leq a$, then $a = b$;
- If $a \leq b$ and $b \leq c$, then $a \leq c$.

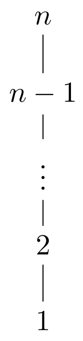
We will use $a < b$ to mean $a \leq b$ and $a \neq b$. We will also write $a > b$ if $b < a$.

Definition 5.3. We say b *covers* a if $a < b$ and there is no c with $a < c < b$. We denote this $a \lessdot b$.

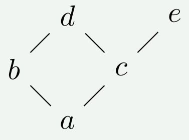
We can represent any finite poset by drawing the directed graph depicting all covering relations. This is called the *Hasse diagram* of P.

Remark 5.4. This isn't true for infinite posets: $(\mathbb{R}, \leq)$ has no covering relations but is a poset.

For example, $[n]$ is a poset, whose Hasse diagram is a chain:



Example 5.5
The following is the Hasse diagram of a poset:



Definition 5.6. A *chain* in a poset P is a sequence

$$a_1 < a_2 < \cdots < a_k.$$

Definition 5.7. A *saturated chain* in a poset P is a chain where

$$a_1 \lessdot a_2 \lessdot \cdots \lessdot a_k.$$

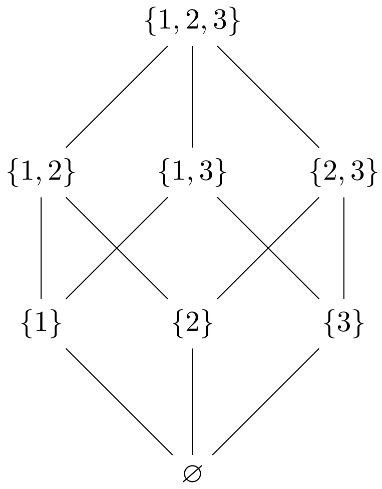
Definition 5.8. An *antichain* is a subset $A \subset P$ such that for all $a \neq b \in A$, a and b are incomparable (meaning that neither $a \leq b$ nor $b \leq a$).

For example, in our second poset, (a, b, d) is a chain and (b, c) is an antichain.

§5.2 Sperner's Theorem

Definition 5.9. A *Boolean lattice* is a poset $B = (2^{[n]}, \subseteq)$ – the set of subsets of $[n]$ under inclusion.

For example, B_3 has a Hasse diagram which is a 3-dimensional cube:



If a poset has a unique minimum, we usually denote it as $\hat{0}$. Similarly, a unique maximum is denoted as $\hat{1}$.

The largest size of a chain in B_n is $n+1$, and Sperner's Theorem is about the largest size of an antichain.

The Boolean lattice has the property of being subdivided into layers:

Definition 5.10. P is a *ranked poset* if there exists a function $\rho : P \to \mathbb{Z}$ such that for all $a \lessdot b$, we have $\rho(b) = \rho(a) + 1$.

In order to not have infinitely many rank functions, we require that the minimal rank is 0.

The Boolean lattice is ranked by cardinality. Not all posets are ranked: for example, the poset whose Hasse diagram is a pentagon is not ranked. In general, a poset is ranked iff for all a and b, any two saturated chains from a to b have the same length.

Definition 5.11. If P is a finite ranked poset, its *rank numbers* are the number of elements of a given rank: the r_i are defined as

$$r_i = \#\{a \in P \mid \rho(a) = i\}.$$

For example, in the Boolean lattice, $r_k = \binom{n}{k}$.

There are a few nice properties ranked posets may have:

Definition 5.12. If P is a finite poset with maximal rank ℓ, then:

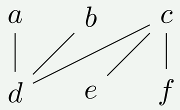
- P is *rank-symmetric* if $r_i = r_{\ell-i}$ for all $0 \leq i \leq \ell$.
- P is *rank-unimodal* if the rank numbers weakly increase and then decrease: there is some k for which
$$r_0 \leq r_1 \leq \cdots \leq r_k \geq r_{k+1} \geq \cdots \geq r_\ell.$$
- P is *Sperner* if the maximal size of an antichain is the maximal rank number.

Note that all elements of a given rank form an antichain, which means the maximal size of an antichain is always *at least* the maximal rank number.

Then Sperner's Theorem states that the Boolean lattice is Sperner.

Example 5.13

The following poset is rank-symmetric and rank-unimodal, but not Sperner:



Here the maximal rank number is 3, but $\{a, b, e, f\}$ is an antichain of size 4.

Definition 5.14. If P is a finite ranked poset with maximal rank ℓ, then a *symmetric chain decomposition* (SCD) of P is a way to decompose P as a disjoint union of saturated chains
$$P = C_1 \sqcup \cdots \sqcup C_k,$$
such that if a chain starts with a_0 and ends with a_s, then $\rho(a_0) = \ell - \rho(a_s)$.

Lemma 5.15

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

Proof. The first two properties are clear. For the third, an antichain can have at most one element from each chain, so if there are k chains, then $|A| \leq k$. But each chain must have one element with rank $\left\lfloor \frac{\ell}{2} \right\rfloor$, so then $k = r_{\lfloor \ell/2 \rfloor}$ is the maximal rank number. □

Theorem 5.16 (de Bruijn 1948)

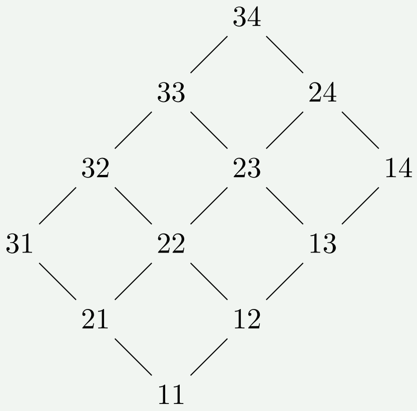
B_n has a symmetric chain decomposition.

If we can show this, then Sperner's Theorem immediately follows. We will prove a more general statement.

Definition 5.17. Given two posets P and Q, $P \times Q$ is the set of all pairs (a, b) with $a \in P$ and $b \in Q$, with the order relation $(a, b) \leq (a', b')$ iff $a \leq a'$ and $b \leq b'$.

Example 5.18

$[m] \times [n]$ is a $m \times n$ triangle rotated by 45°. For example, $[3] \times [4]$ is the following:



Then the Boolean lattice is $[2] \times [2] \times \cdots \times [2]$.

Theorem 5.19

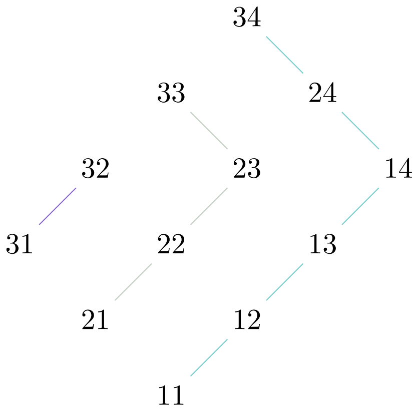
Any product of chains $[a_1] \times \cdots \times [a_n]$ has a symmetric chain decomposition.

This theorem is more general than Sperner's Theorem, but it's actually easier to prove, since we can do induction. It follows from two simple lemmas:

Lemma 5.20

The product of two chains $[a] \times [b]$ has a symmetric chain decomposition.

Proof. Take L-shaped paths on the rectangle:



□

Lemma 5.21

If P and Q have symmetric chain decompositions, then so does $P \times Q$.

Proof. Take a SCD $P = C_1 \cup \cdots \cup C_k$, and $Q = D_1 \cup \cdots \cup D_\ell$. Then as a set, we have

$$P \times Q = \bigsqcup C_i \times D_j.$$

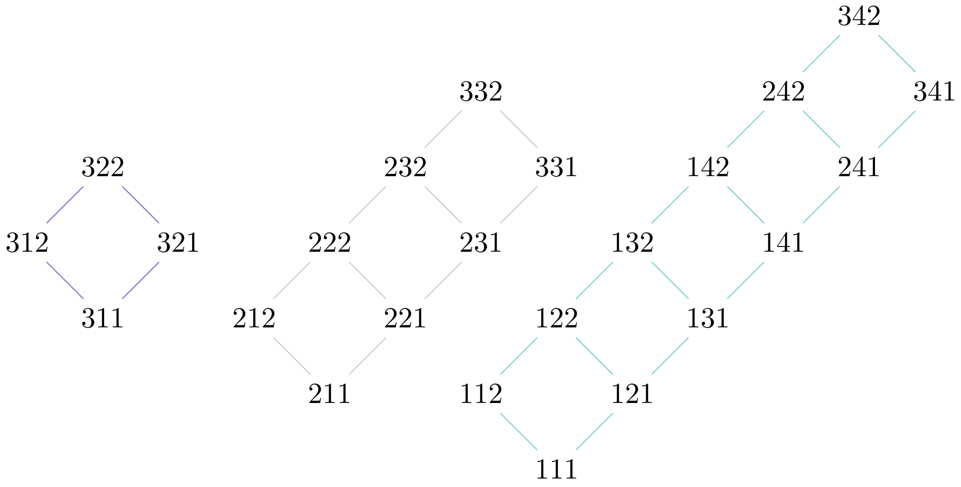
We can pick SCDs for each $C_i \times D_j$ by the first lemma, and combining them gives a SCD of $P \times Q$. □

Now to prove that $[a_1] \times \cdots \times [a_n]$ has a symmetric chain decomposition, we can simply induct on n.

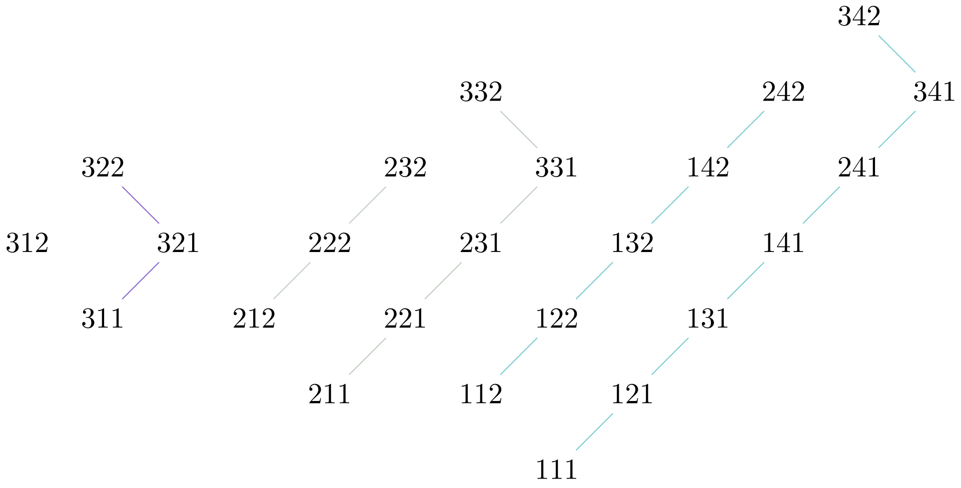
Example 5.22

Construct a SCD of $[3] \times [4] \times [2]$.

Proof. We can take the above SCD of $[3] \times [4]$, and then "straighten out" each of our chains (and ignore the edges outside the chains). Then each chain gets made two-dimensional:



Now we can take the same L-shaped construction on each of these rectangles:



This gives us a SCD of $[3] \times [4] \times [2]$. □

§5.3 Chains and Antichains

Let P be any (not necessarily ranked) finite poset.

Definition 5.23. Let $\mathrm{MC}(P)$ and $\mathrm{MA}(P)$ be the maximal sizes of a chain and antichain, respectively. Let $\mathrm{ma}(P)$ and $\mathrm{mc}(P)$ be the minimal number of antichains and chains needed to cover all of P, respectively.

Clearly $\mathrm{ma}(P) \geq \mathrm{MC}(P)$, since every element in the maximal chain must come from a different antichain. But it turns out much more is true:

Theorem 5.24 (Dilworth 1950)

For any finite poset, $\mathrm{MC}(P) = \mathrm{ma}(P)$.

Theorem 5.25 (Mirsky 1971)

For any finite poset, $\mathrm{MA}(P) = \mathrm{mc}(P)$.

There actually exists a generalization of both results: let P be a finite poset with n elements. For $k \geq 1$, define ℓ_k to be the maximal size of a union of k chains in P, and m_k the maximal size of a union of k antichains.

Then $\ell_1 = \mathrm{MC}$ and $m_1 = \mathrm{MA}$, and it's clear that both sequences are weakly increasing and eventually stabilize at n – the ℓ_i stabilize at mc, and the m_i stabilize at ma.

Theorem 5.26 (Greene 1970)

Consider the sequences

$$\lambda(P) = (\ell_1, \ell_2 - \ell_1, \ell_3 - \ell_2, \ldots) = (\lambda_1, \lambda_2, \ldots)$$

of the consecutive differences of ℓ_i, and

$$\mu(P) = (m_1, m_2 - m_1, m_3 - m_2, \ldots) = (\mu_1, \mu_2, \ldots)$$

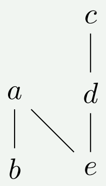
of the consecutive differences of m_i.

Then $\lambda(P)$ and $\mu(P)$ are partitions of n which are conjugate to each other.

In particular, the condition that they're partitions implies $\lambda_1 \geq \lambda_1 \geq \cdots$ and $\mu_1 \geq \mu_2 \geq \cdots$. Conjugation of a partition corresponds to flipping the Young diagram over the main diagonal. We can write conjugate partitions as $\mu = \lambda'$.

Example 5.27

Construct λ and μ for the following poset:



Solution. The longest chain has length 3 (cde), and we can cover the entire poset with 2 chains, so the ℓ_i are $(3, 5, 5, \ldots)$. Similarly, the m_i are $(2, 4, 5, 5, \ldots)$. So then the partitions are $\lambda = (3, 2)$ and $\mu = (2, 2, 1)$, which are conjugate. □

Dilworth's Theorem states that the first row of λ is the first column of μ, and Mirsky's Theorem states that the first row of μ is the first column of λ. So both are implied by Greene's Theorem.

Greene's Theorem is hard to prove, and we won't prove it in this class. But it has special implications in certain kinds of posets.

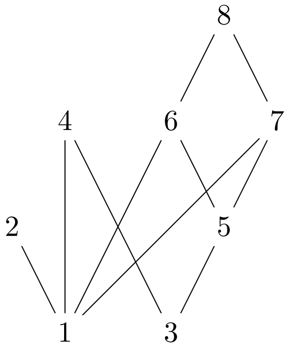
§5.3.1 Permutation Posets

Definition 5.28. Given a permutation $w = w_1 \ldots w_n \in S_n$, its *permutation poset* is the poset P_w on $\{1, 2 \ldots, n\}$ such that $w_i \leq_P w_j$ if $w_i \leq w_j$ and $i \leq j$.

Example 5.29

Draw the permutation poset of $w = 35176842$.

Solution. We rank pairs of elements which occur in the correct order in w:



□

Then chains in P_w are in exact correspondence with increasing subsequences (for example, 3578); similarly antichains are in exact correspondence with decreasing subsequences (for example, 2467).

There are many questions people ask about increasing and decreasing subsequences in permutations. We can use Greene's Theorem to define the partitions λ and μ for these posets. We call λ the *shape* of the permutation.

Example 5.30

Find the shape of our above example $w = 35176842$.

Solution. The maximal chain has length 4 (3578), the maximal union of two chains has size 6 (3578 and 16), the maximal union of three chains has size 7, and we can cover all elements with four chains. So then $\lambda = (4, 2, 2, 1)$. □

Corollary 5.31 (Erdos-Szekeres)

Let m and n be positive integers. Then any permutation of size at least $mn+1$ must have an increasing subsequence of size $m+1$ or a decreasing subsequence of size $n+1$.

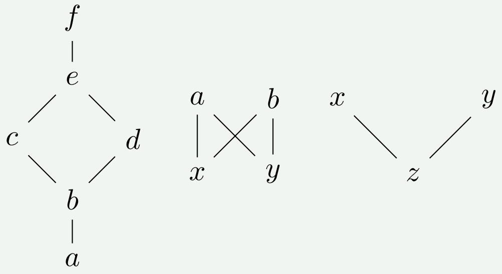
Proof. Assume not. Then the partition has a Young diagram whose first row is at most m, and whose column is at most n. So the Young diagram fits in a $m \times n$ rectangle and has at most mn boxes, contradiction. □

§5.4 Lattices

Definition 5.32. Let L be a poset. For any $x, y \in L$, we can define x *join* y, denoted $z = x \vee y$, as the unique minimal element of L such that $z \geq x$ and $z \geq y$ (if it exists). Similarly, we can define x *meet* y, denoted $t = x \wedge y$, as the unique maximal element of L such that $t \leq x$ and $t \leq y$. A *lattice* is any poset such that for any x and y in L, there exists $x \vee y$ and $x \wedge y$.

Example 5.33

Consider the following three posets:



The first is a lattice. The second is not a lattice because $x \wedge y$ and $x \vee y$ are both not defined (no vertex satisfies $z \leq x, y$, and both a and b satisfy $z \geq x, y$). The third is not a lattice as $x \vee y$ does not exist, but it is a *meet-semilattice*, where meet always exists.

It is also possible to define lattices axiomatically:

Definition 5.34. A set L with binary operations $\vee$ and $\wedge$ is a lattice if it satisfies the following properties:

- Commutativity: $x \vee y = y \vee x$ and $x \wedge y = y \wedge x$.
- Associativity: $(x \vee y) \vee z = x \vee (y \vee z)$ and $(x \wedge y) \wedge z = x \wedge (y \wedge z)$.
- The Absorption Law: $x \vee (x \wedge y) = x$, and $x \wedge (x \vee y) = x$.

Proposition 5.35

These definitions are cryptomorphic, meaning they define the same object.

Proof. Given a poset in the first definition, we can easily check that the axioms hold. On the other hand, to go from the second definition to the first, take $x \leq y$ iff $x \vee y = y$. Note that the second half of the Absorption Law implies that if $x \vee y = y$, then $x \wedge y = x$; similarly the first half implies the opposite direction. So these are equivalent.

Then we can check that the axioms of a lattice imply the axioms of a poset:

- To show $x \leq x$, we have
$$x \vee x = x \vee (x \wedge (x \vee y)) = x.$$
- If we have $x \leq y$ and $y \leq x$, then $x \vee y$ must equal both x and y (by commutativity), so $x = y$.
- If $x \leq y$ and $y \leq z$, then $x \vee y = y$ and $y \vee z = z$. This means $x \wedge y = x$ and $y \wedge z = y$, so
$$z \vee x = z \vee (y \wedge x) = z \vee ((z \wedge y) \wedge x) = z \vee (z \wedge (y \wedge x)) = z,$$
which means $x \leq z$.

Now it suffices to check that $x \vee y$ is the unique minimal z such that $x \wedge z = x$ and $y \wedge z = y$. First, $x \vee y$ satisfies these equations by the Absorption Law. On the other hand, suppose z satisfies both equations. Then we have
$$z \vee (x \vee y) = (z \vee x) \vee y = z \vee y = z,$$
so $z \geq x \vee y$. □

Example 5.36

A chain is a lattice, and a product of chains $[n] \times [m]$ is also a lattice (where meet and join are the bottom and top corners of the rectangle formed by x and y).

The Boolean lattice B_n is a lattice, where $x \vee y$ is the union of x and y, and $x \wedge y$ is the intersection.

We can think of meet and join as generalizations of intersection and union.

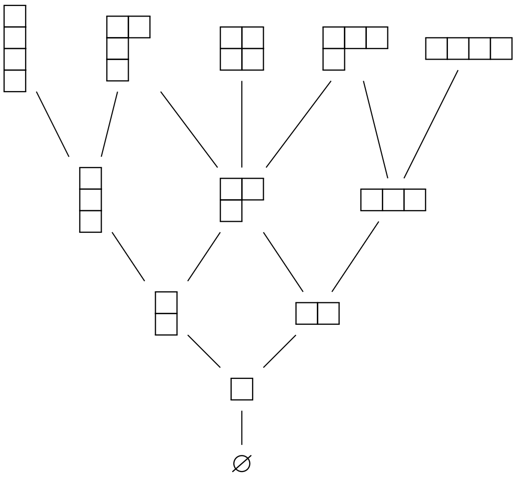
§5.4.1 Young's Lattice

Definition 5.37. *Young's Lattice* $\mathbb{Y}$ is the poset of all Young diagrams, ordered by inclusion.

This is an infinite poset – it contains all Young diagrams of all sizes. Its unique minimal element is $\varnothing$.

The covering relation is that $\lambda \lessdot \mu$ if λ is μ with one box removed.

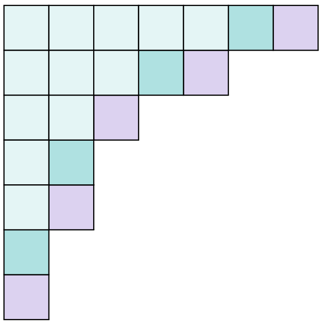
We can draw the first few rows as follows:



Proposition 5.38

The number of edges above a Young diagram λ is exactly one more than the number of edges below λ.

Proof. To go down, we delete a corner. To go up, we add a box at an *outer* corner:



Here the original diagram is in blue, the boxes which can be deleted are in darker blue, and the boxes which can be added are in red. If we trace the border of the Young diagram (on the bottom-right), adding an extra edge at the beginning and end, then the places we can add a box are exactly the outward-facing turns (or valleys), and the places we can delete a box are exactly the inward-facing turns (or peaks). We alternate between them, starting and ending with a valley, so there is exactly one more place where we can add a box than place where we can delete a box. □

Proposition 5.39

$\mathbb{Y}$ is a lattice.

Proof. If we place two lattices on top of each other (aligned at their top-left corner), then meet is their set-theoretic intersection, and join is their set-theoretic union. □

§5.4.2 Order Ideals

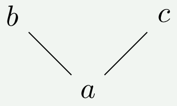
Definition 5.40. Let P be any poset. Then a subset $I \subset P$ is called a *order ideal* if for all $x \in I$ and $y \leq x$ in P, we have $y \in I$.

So I is closed downwards: if $x \in I$, everything below it is in I as well.

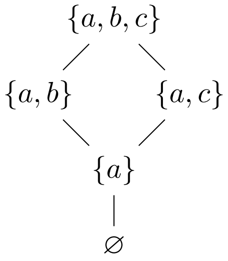
Definition 5.41. Given a poset P, its *lattice of order ideals* $J(P)$ is the poset of all its order ideals, ordered by inclusion.

Example 5.42

Find the lattice of order ideals of the following poset:



Solution. We can write out all order ideals, and we get the following Hasse diagram:



□

The unique minimum is $\hat{0} = \varnothing$, and the unique maximum $\hat{1}$ is the entire set.

> **Remark 5.43.** Every finite lattice has a unique minimal and maximal element, since if there were two maximal elements, their join could not exist.

Lemma 5.44

$J(P)$ is a lattice.

Proof. We have $I \vee J = I \cup J$, and $I \wedge J = I \cap J$. (It's easy to check these are order ideals.) □

Example 5.45

B_n is a lattice of order ideals.

Proof. Take P to be the "empty" poset consisting of n points with no order relations. Then any subset is an order ideal, so $J(P)$ is the set of all subsets. □

Example 5.46

Describe the lattice of order ideals for the chains $[n]$ and $\mathbb{Z}_{\geq 0}$.

Proof. We have

$$J([n]) = \{\varnothing, \{1\}, \{1,2\}, \dots, \{1,2,\dots,n\}\} \cong [n+1].$$

For the same reason, $J(\mathbb{Z}_{\geq 0}) \cong \mathbb{Z}_{\geq 0}$. □

Lemma 5.47

The lattice of order ideals of $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is $\mathbb{Y}$.

Proof. The Hasse diagram of $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is a square grid (the first quadrant of the coordinate plane) rotated 45°. If we treat the dots as boxes, then order ideals are exactly Young diagrams. □

Definition 5.48. A *distributive lattice* $(L, \wedge, \vee)$ is a lattice satisfying the Distributive Law:

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

and analogously

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

Note that unlike with addition and multiplication – where it's not true that $x + yz = (x + y)(x + z)$ – the Distributive Law works both ways.

Theorem 5.49 (Fundamental Theorem on Finite Distributive Lattices)

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

Conversely, for any finite distributive lattice L, there exists a finite poset P such that $L \cong J(P)$.

This is also known as Birkhoff's Representation Theorem.

For the first part, it's easy to check that the distributive laws hold for the union and intersection of sets. (This part is true even if the poset is not finite.)

For the second, let L be any finite distributive lattice. We say $x \in L$ is *join-irreducible* if $x \neq \hat{0}$, and there are no two $y, z \in L$ with $y, z < x$, such that $x = y \vee z$.

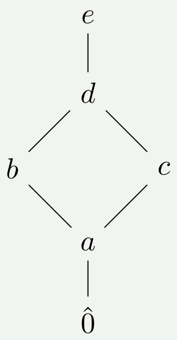
Lemma 5.50

Let P be the sub-poset of L given by all join-irreducible elements. Then $L \cong J(P)$.

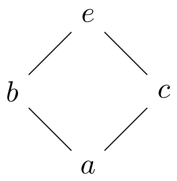
We won't prove it here, but it's possible to prove this from the axioms of distributive lattices.

Example 5.51

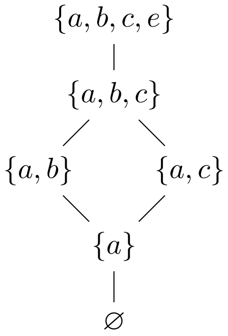
Demonstrate this for the following lattice L:



Solution. The join-irreducible elements are a, b, c, and e. So P is a diamond:



The order ideals are $\varnothing$, $\{a\}$, $\{a, b\}$, $\{a, c\}$, $\{a, b, c\}$, and the entire set. So $J(P)$ is the following:



We can see this is identical to the original lattice. □

§5.5 Young's Lattice Again

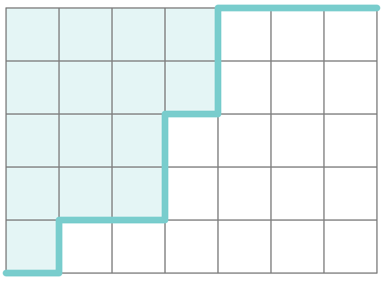
We saw $J(\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}) = \mathbb{Y}$. We can also take a product of two *finite* chains: $J([m] \times [n])$ is the poset of Young diagrams which fit inside a $m \times n$ rectangle. We call this $L(m, n)$.

Definition 5.52. $L(m, n)$ is the sublattice of $\mathbb{Y}$ formed by all Young diagrams $\lambda \subseteq m \times n$.

Proposition 5.53

The size of $L(m, n)$ is $\binom{m+n}{n}$.

Proof. Young diagrams which fit inside the box biject to lattice paths between opposite corners:

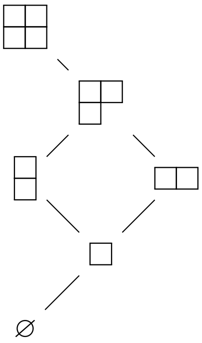


There are $\binom{m+n}{n}$ such paths, so $\binom{m+n}{n}$ Young diagrams as well. □

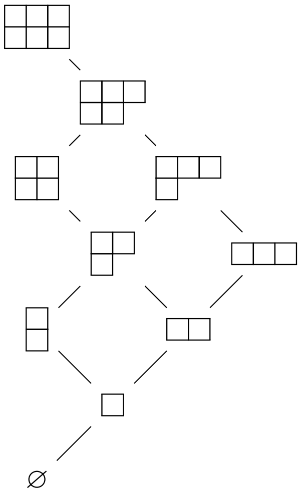
Example 5.54

Find $L(2, 2)$ and $L(2, 3)$.

Solution. $L(2, 2)$ is the Young diagrams which fit inside a 2×2 square:



Similarly, $L(2,3)$ is the Young diagrams which fit inside a 2×3 rectangle:



These look similar (they are half of a square grid), and we can show $L(2,n)$ always looks like this. □

Definition 5.55. A *linear extension* of a finite poset P is a labelling of the elements of P with 1 through n (without repeitition) such that for any $x < y$ in P, the label of x is less than the label of y.

Proposition 5.56

The number of saturated chains from $\hat{0}$ to $\hat{1}$ in $J(P)$ is the number of linear extensions of P.

Proof. This follows directly from the definitions. If $I_1 \lessdot I_2$, then I_2 must be I_1 with one element added (otherwise consider adding just the minimal new element), so saturated paths correspond exactly to labellings (where we label elements in the order we add them). □

This is often a convenient way of looking at things. For example, the linear extensions of $L(2,3)$ are Standard Young Tableaux whose shape is a 2×3 rectangle – we can convert each tableaux into a saturated chain by adding boxes in order.

This gives a correspondence between SYT whose shape is a $2 \times n$ rectangle and Dyck paths, which we've already seen. But the point is that we can think of SYT as saturated chains in a lattice.

Theorem 5.57

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

Proof of rank-symmetry. If we place a Young diagram with k boxes in a $m \times n$ rectangle, then its complement is a Young diagram with $mn - k$ boxes (rotated 180°). □

Rank-symmetry is fairly easy to prove. The others are much harder. Rank-unimodality was first proved by Sylvester in 1878, and the proof he gave was nonconstructive – the first constructive proof was given by O'Hara in the 1990s. Meanwhile, the Sperner property was proved by Stanley in 1980.

For example, the rank numbers of $L(2, 3)$ are 1, 1, 2, 2, 2, 1, 1.

§5.5.1 q-analogs

It turns out that the rank numbers of $L(m, n)$ are special: they are related to q-binomial coefficients.

Suppose we have a classical object (for example, binomial coefficients). Then its q-analog is a polynomial in q with positive integer coefficients, whose value at $q = 1$ is our classical object.

This is somewhat similar to the idea of statistics discussed earlier.

Definition 5.58. The q-analog of a number is

$$[n]_q := 1 + q + q^2 + \cdots + q^{n-1} = \frac{1 - q^n}{1 - q}.$$

Definition 5.59. The q-factorial is

$$[n]_q! := [1]_q[2]_q[3]_q \cdots [n]_q.$$

Definition 5.60. The q-binomial coefficient is

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

A priori we only know this is a rational function, but it turns out that it's actually an integer coefficient polynomial, and its coefficients are rank numbers.

Example 5.61

Calculate $\left[\begin{smallmatrix} 4 \\ 2 \end{smallmatrix}\right]_q$.

Proof. We get

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

We can see that this is a polynomial with positive integer coefficients, which are symmetric and unimodal. □

Theorem 5.62

We have

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

Proof. Use induction; then it suffices to show that

$$\sum_{w \in S_n} q^{\mathrm{inv}(w)} = \sum_{u \in S_{n-1}} q^{\mathrm{inv}(u)} \cdot (1 + q + \cdots + q^{n-1}).$$

But we can obtain permutations of $[n]$ by taking permutations of $[n-1]$ and inserting n. If we insert n in the end, this adds 0 inversions; if we add it before u_{n-1}, this adds 1 inversion; and so on. So each choice adds $0, 1, \ldots, n-1$ inversions to any $u \in S_{n-1}$, which corresponds exactly to multiplication by $[n]_q$. □

Theorem 5.63

We have

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

It is possible to prove this theorem using recurrence relations. The idea is to find a recurrence for q-binomial coefficients, and show that the same recurrence holds for the right-hand side.

Lemma 5.64 (q-Pascal's Recurrence)

We have

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

Proof. This is easy to see from the definitions: we want to show

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

Cancelling common factors, this reduces to

$$[n]_q = [k] + q^k[n-k],$$

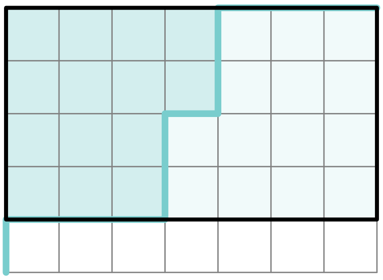
which is true. □

Proof of Theorem 5.63. It suffices to show that the same recurrence holds for

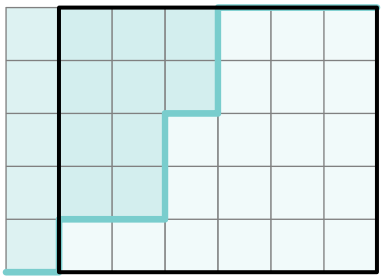
$$R(n,k) = \sum_{\lambda \subseteq k \times (n-k)} q^{|\lambda|}$$

as well, meaning that $R(n,k) = R(n-1,k-1) + q^k R(n-1,k)$.

We can think of Young diagrams $\lambda \subseteq k \times (n-k)$ as lattice paths from the lower-left to top-right corner.



If the first step is vertical, then the rest of the lattice path is a lattice path in a $(k-1) \times (n-k)$ rectangle, contributing the first term.



Meanwhile, if the first step is horizontal, then the rest of the path is also a lattice path in a $(n-k-1) \times k$ rectangle. But the first column also contributes q^k boxes. So this gives the second term.

Then the left-hand side and right-hand side satisfy the same recurrence, so must be equal. □

§5.5.2 Grassmannians

We saw the identity

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

We proved this via recursion, but there is a more conceptual proof, involving some underlying geometry.

Definition 5.65. For a field $\mathbb{F}$, the *Grassmannian* $\mathrm{Gr}(k, n; \mathbb{F})$ is the set of k-dimensional linear subspaces in $\mathbb{F}^n$.

Example 5.66

Describe $\mathrm{Gr}(1, 2; \mathbb{R})$.

Solution. A point in the Grassmannian is a line in $\mathbb{R}^2$ passing through the origin. This Grassmannian is known as the *projective line.*

We can think of this Grassmannian by drawing a circle around the origin; every line in the Grassmannian intersects the circle at two opposite points, so we can identify lines in the Grassmannian by gluing opposite points on a circle together. □

We can describe Grassmannians more concretely in terms of matrices. Given a k-dimensional subspace of $\mathbb{F}^n$, we can pick a collection of k basis vectors $v_1, v_2, \ldots, v_k$ in $\mathbb{F}^n$. Then we can put these vectors into a

$k \times n$ matrix of rank k:

$$A = \begin{bmatrix} - & v_1 & - \\ & \vdots & \\ - & v_k & - \end{bmatrix}.$$

Every matrix of this form represents a point on the Grassmannian. But we're not really interested in the matrix itself, only the space spanned by the rows. So the Grassmannian is the space of $k \times n$ matrices of rank k, modulo row operations.

Example 5.67

Describe $\operatorname{Gr}(1, 2; \mathbb{R})$ again, this time in terms of matrices.

Solution. We consider 1×2 matrices (which are not 0). Then row operations just rescale the matrix. So if the first entry is nonzero, then we can rescale it to 1 – this gives us $\{(1, x)\} \cup \{(0, 1)\}$, which is a line plus a point "at infinity". □

Now assume q is a prime power, and take $\mathbb{F}$ to be the finite field $\mathbb{F}_q$ with q elements (which exists for all prime powers q).

The idea is to count $\# \operatorname{Gr}(k, n; \mathbb{F}_q)$ in two ways. (It suffices to prove our identity for prime powers q, since it is a polynomial identity in q.)

Theorem 5.68

We have

$$\# \operatorname{Gr}(k, n; \mathbb{F}_q) = \begin{bmatrix} n \\ k \end{bmatrix}_q.$$

Proof. The Grassmannian is the space of $k \times n$ matrices of rank k, modulo row operations. Performing a series of row operations corresponds to multiplying by an invertible matrix on the left – so we want to identify our matrices up to left multiplication by invertible $k \times k$ matrices. This means

$$\# \operatorname{Gr}(k, n) = \frac{\#\{k \times n \text{ matrices of rank } k\}}{\#\{\text{invertible } k \times k \text{ matrices}\}}.$$

Now to find the numerator, imagine picking the rows one by one. First, v_1 can be any nonzero vector, giving $q^n - 1$ choices. Then v_2 can be any vector not in $\operatorname{Span}(v_1)$, giving $q^n - q$ ways. Then v_3 can be any vector not in $\operatorname{Span}(v_1, v_2)$, giving $q^n - q^2$ ways, and so on. So the number of $k \times n$ matrices of rank k is

$$(q^n - 1)(q^n - q)(q^n - q^2) \cdots (q^n - q^{k-1}).$$

For the same reason, the denominator is

$$(q^k - 1)(q^k - q)(q^k - q^2) \cdots (q^k - q^{k-1}).$$

So then after cancelling out powers of q, and cancelling $q - 1$ from each term, we get

$$\# \operatorname{Gr}(k, n) = \frac{(q^n - 1) \cdots (q^n - q^{k-1})}{(q^k - 1) \cdots (q^k - q^{k-1})} = \begin{bmatrix} n \\ k \end{bmatrix}_q.$$

□

On the other hand, there's another way to count the number of elements in the Grassmannian: the idea is to think about Gaussian elimination. There is a canonical form of thinking about a matrix up to row operations:

Lemma 5.69

Given a $k \times n$ matrix A, by performing row operations, we can transform A uniquely to a matrix $\tilde{A}$ in *reduced row echelon form.*

Proof. Essentially, we start looking at columns from the left, until we find one with a nonzero entry. Then we move that entry to the top, rescale it to 1, and use it to kill all entries below it. Now proceed looking at columns until we find one with a nonzero entry not in the first row. Move it to the second row, rescale it to 1, and use it to kill everything else in this column (which doesn't affect columns to our left, since this row necessarily has all zeroes to the left of this 1). Now proceed looking until we find a column with a nonzero entry not in the first or second row, and so on.

This produces a matrix with some 1s, which move strictly to the right as we go down. In the columns of the 1s, all other entries are 0; otherwise we can have any entries $*$ to the right of the 1s:

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

The 1's are called *pivots*, and the columns containing the 1's are called *pivot columns.* □

The number of pivots is the rank of the matrix, so our matrices must all have k pivots.

Proposition 5.70

We have

$$\#\operatorname{Gr}(k, n; \mathbb{F}_q) = \sum_{\lambda \subseteq k \times (n-k)} q^{|\lambda|}.$$

Proof. Count Grassmannians by the positions of pivots. Then if we have s stars, there are q^s Grassmannians, since each star has q possibilities.

Now imagine deleting all pivot columns. Then we get a matrix with k rows and $n - k$ columns, and the $*$ entries form a reflected Young diagram:

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

On the other hand, given the Young diagram we can uniquely recover the pivot columns: the pivot columns are exactly where we gain one row. So each Young diagram $\lambda \subseteq k \times (n - k)$ contributes $q^{|\lambda|}$ matrices. □

Remark 5.71. This proof shows where Young diagrams are coming from in our identity: they come from row echelon forms. In fact, the decomposition of Grassmannians based on their Young diagram is called the *Schubert decomposition.*

§5.5.3 More on q-analogs

We have already seen the formula

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

Meanwhile, there is a similar interpretation of the identity we just proved: let u be a permutation of the multiset of k 1's and $n-k$ 2's. Then our identity becomes

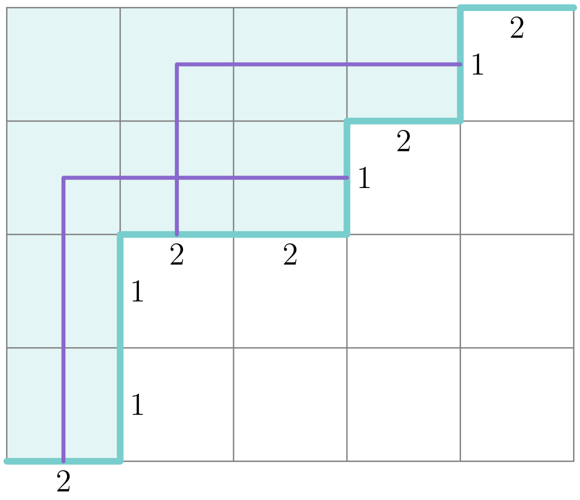
$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \sum_u q^{\text{inv}(u)},$$

where $\text{inv}(u)$ is similarly defined as the number of pairs (i,j) with $i<j$ and $u_i > u_j$:

Proposition 5.72

There is a bijection between Young diagrams $\lambda \subseteq k \times (n-k)$ and permutations u of the multiset with k 1's and $n-k$ 2's, such that $|\lambda| = \text{inv}(u)$.

Proof. Read u as the instructions for a path, where 2 represents a right-step and 1 an up-step. For example, $u = 211221212$ (which has 9 inversions) gives the following path and Young diagram:



There is a bijection between inversions and boxes of λ: each 2 before a 1 corresponds to a box in the 2's column and 1's row. So $|\lambda| = \text{inv}(u)$. □

We can generalize both results:

Definition 5.73. If $k_1 + \cdots + k_\ell = n$, we define the multinomial coefficient

$$\binom{n}{k_1, \ldots, k_\ell} := \frac{n!}{k_1! \cdots k_\ell!}.$$

This is the number of permutations of the multiset with k_i i's for each $1 \le i \le \ell$.

Definition 5.74. The q-multinomial coefficient is defined as

$$\begin{bmatrix} n \\ k_1, \ldots, k_\ell \end{bmatrix}_q := \frac{[n]_q!}{[k_1]_q! \cdots [k_\ell]_q!}.$$

Theorem 5.75

We have

$$\begin{bmatrix} n \\ k_1, \dots, k_\ell \end{bmatrix}_q = \sum_u q^{\mathrm{inv}(u)},$$

where we sum u over all permutations of the multiset with k_1 1's, k_2 2's, and so on.

We can still prove this by induction. The corresponding geometric argument would be to use flag varieties instead of Grassmannians.

§6 Young Tableaux Again

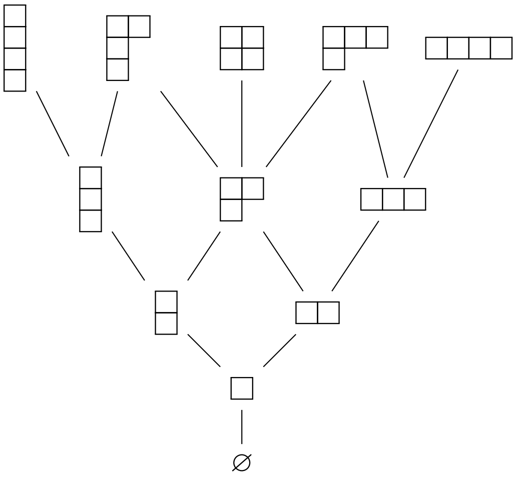
Earlier, we defined f_λ as the number of Standard Young Tableaux of shape λ. We found an explicit formula for f_λ, the Hook Length Formula. But there are other things we can do with f_λ as well:

Theorem 6.1

For all n, we have

$$\sum_{\lambda \vdash n} (f_\lambda)^2 = n!$$

(where the sum is over all partitions λ of n).



One way to think about the f_λ is in terms of Young's Lattice $\mathbb{Y}$. Each λ is one node of the lattice, and f_λ is the number of saturated chains from $\varnothing$ to λ – we can imagine building λ in the order 1, 2, . . . , n.

For example, when $n = 4$, the theorem states that

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

> **Remark 6.2.** This theorem has relevance to representation theory. Given any finite group, if its irreducible representations have dimensions d_i, then the order of the group is $\sum d_i^2$. In S_n, irreducible representations are labelled by Young diagrams with n boxes, and f_λ is the dimension of that irreducible representation.

One way to prove this is via Schensted correspondence:

Proposition 6.3 (Schensted Correspondence)

There is a bijection between S_n and the set of pairs (P, Q) of SYT's of the same shape $\lambda \vdash n$.

This bijection would prove the identity; we will discuss it later.

§6.1 Up and Down Operators

Another way to think about the identity is in terms of *up* and *down* operators.

Consider $\mathbb{R}[\mathbb{Y}]$, the space of formal linear combinations of Young diagrams. For example, elements of $\mathbb{R}[\mathbb{Y}]$ can look like

$$6 \cdot \ydiagram{2,1,1} - 2 \cdot \ydiagram{3,2} + \pi \cdot \ydiagram{2,1}.$$

This is an infinite-dimensional vector space, with a basis given by single Young diagrams.

Now we define two linear operators acting on $\mathbb{R}[\mathbb{Y}]$ (which can be thought of as infinite matrices with rows and columns labelled by Young diagrams). It suffices to define what the operators do to the basis elements.

Definition 6.4. The up-operator is defined as

$$U : \lambda \mapsto \sum_{\mu \gtrdot \lambda} \mu,$$

and similarly the down-operator is defined as

$$D : \lambda \mapsto \sum_{\mu \lessdot \lambda} \mu.$$

So the up-operator sends λ to the sum of all Young diagrams obtained by adding a box to λ, and the dow-operator sends λ to the sum of all Young diagrams obtained by removing a box.

We can compose operations:

Example 6.5

Find $DU(\lambda)$ and $UD(\lambda)$ for $\lambda = \ydiagram{2}$.

Solution. We have

$$DU\left(\ydiagram{2}\right) = D\left(\ydiagram{3} + \ydiagram{2,1}\right) = 2 \cdot \ydiagram{2} + \ydiagram{1,1}.$$

Similarly, we can calculate that

$$UD\left(\ydiagram{2}\right) = U\left(\ydiagram{1}\right) = \ydiagram{2} + \ydiagram{1,1}.$$

□

Notice that in the above example, $(DU - UD)(\lambda) = \lambda$. This is not a coincidence:

Lemma 6.6

$[D, U] = DU - UD$ (called the *commutator* of D and U) is the identity.

Proof. Let the coefficient of μ in $[D, U](\lambda)$ be $c_{\lambda\mu}$.

Case 1 ($\lambda \neq \mu$). Then when we apply DU, we start with λ, add a box, and remove a *different* box to get μ. Meanwhile, for UD, we remove a box and then add a different box. Since the boxes are different, the two orders cancel out, which means $c_{\lambda\mu} = 0$.

Case 2 ($\lambda = \mu$). Then when we apply DU, we add a box and remove the same box, while when we apply UD, we remove a box and add the same box. So the coefficients in DU and UD are the number of ways to add a box and remove a box, respectively.

But we saw earlier that there's exactly one more way to add a box than to remove a box: these correspond to outward-facing and inward-facing corners on the border of λ, which alternate as we walk along the border. So if λ has c outer corners (meaning corners which we can delete), we get

$$c_{\lambda\mu} = (c+1) - c = 1.$$

So then $[D, U](\lambda) = \lambda$ for all Young diagrams λ, which means $[D, U]$ is the identity. □

The key idea is that if $\lambda \in \mathbb{Y}$ covers c diagrams, then $c+1$ diagrams cover it.

§6.1.1 Differential Posets

> **Definition 6.7.** A ranked poset P with a unique minimal element $\hat{0}$ is called a *differential poset* if its up and down operators satisfy $[D, U] = I$.

The up and down operators are defined similarly, as linear operators on $\mathbb{R}[P]$ sending $x \in P$ to the sum of all nodes covering it and the sum of all nodes covered by it, respectively.

We just proved that $\mathbb{Y}$ is a differential poset.

Combinatorially, a poset is differentiable iff for all $\lambda \neq \mu$, if there are a ways to go up and then down from λ to μ, then there are a ways to go down and then up from λ to μ as well. (In $\mathbb{Y}$, we always have $a = 0$ or 1, but this is not necessarily true in general.) Meanwhile, for each λ, if there are c elements below it, there are $c+1$ elements above.

Then our theorem holds more generally:

Theorem 6.8

For any differential poset P, we have

$$\sum_{\lambda \text{ of rank } n} \#\{\text{saturated chains in } P \text{ from } \hat{0} \text{ to } \lambda\}^2 = n!\,.$$

To prove this, the left-hand side is the coefficient of $\hat{0}$ in $D^n U^n(\hat{0})$ – essentially, given a differential poset, we start at $\hat{0}$, go up n steps, and come down n steps. The left-hand side sums the number of ways to do this by casework on which point λ we reach on the nth level – then we take a chain from $\hat{0}$ to λ, and another back from λ to $\hat{0}$.

Now we have the identities

$$DU = UD + I \text{ and } D(\hat{0}) = 0.$$

(Note that $\hat{0}$ is not 0 – 0 is the zero vector, and $\hat{0}$ is the minimal element of the poset.)

These two identities formally imply that

$$D^n U^n(\hat{0}) = n! \cdot \hat{0},$$

meaning that we can do algebraic manipulations using just those two identities to deduce this.

Example 6.9

Calculate $DU(\hat{0})$ and $D^2U^2(\hat{0})$.

Solution. When $n = 1$, we have

$$DU(\hat{0}) = (UD + I)(\hat{0}) = \hat{0},$$

since the $UD(\hat{0})$ term contributes 0. For $n = 2$ we have

$$DDUU(\hat{0}) = D(UD + I)U(\hat{0}) = DUDU(\hat{0}) + DU(\hat{0}).$$

We already saw $DU(\hat{0}) = \hat{0}$, so this is $2\hat{0}$. □

It is not hard to prove the general case by induction, but there are ways of proving it that explain where $n!$ comes from.

Proof 1 of Theorem 6.8. Consider the expression

$$\underbrace{DD\dots D}_{n}\underbrace{UU\dots U}_{n}(\hat{0}).$$

We can think of the U's as particles and D's as antiparticles. The identity $DU = UD + I$ means that a D can either *jump over* a U (so $DU \mapsto UD$) or *annihilate* a U (meaning that $DU \mapsto I$, so both disappear). If an antiparticle jumps past all of the U's, then we end up with $D(\hat{0})$, which contributes nothing. So we can assume every D bumps into (and annihilates) some U.

For example, we could have the sequence

$$D_3D_2D_1UUU \mapsto D_3D_2UD_1UU \mapsto D_3D_2UU$$

where D_1 jumps over the first U and annihilates the second, and then similarly

$$D_3D_2UU \mapsto D_3UD_2U \mapsto D_3U \mapsto I.$$

We sum over all possible sequences.

But the possible sequences of jumps and annihilations correspond exactly to matchings between the U's and D's (multiplied by $\hat{0}$) – each D is matched to the U it annihilates. There are n U's and n D's, so $n!$ ways to match them. □

There is another proof, which explains why these posets are called differential posets:

Proof 2 of Theorem 6.8. As we saw, our two formulas

$$DU = UD + I \text{ and } D(\hat{0}) = 0$$

imply that $D^nU^n = N(\hat{0})$ for some N – we can keep formally applying the rules to move all the D's to the right. So it suffices to show that $N = n!$.

But we can now take *any* pair of operators that satisfy these two relations: we know that the relations formally imply $D^nU^n = N(\hat{0})$ for a fixed N, so it suffices to calculate N for any pair of operators.

Now take operators acting on the linear space of polynomials $\mathbb{R}[x]$, with

$$\tilde{U} : f(x) \mapsto xf(x) \text{ and } \tilde{D} : f(x) \mapsto f'(x).$$

Here the polynomial 1 corresponds to $\hat{0}$. Then the first identity

$$\tilde{D}\tilde{U} = \tilde{U}\tilde{D} + I \iff (x \cdot f(x))' = x \cdot f'(x) + f(x)$$

is just the Product Rule; meanwhile the second identity is true because the derivative of 1 is 0.

But $D^nU^n(1)$ is the nth derivative of x^n, which we know is $n!$. □

Remark 6.10. These posets are called differential posets because they satisfy the product rule for derivatives, which makes them related to derivatives in some sense.

§6.2 Robinson-Schensted Correspondence

(Guest lecture by Professor Anna Weigandt.)

Let $\lambda = (\lambda_1, \ldots, \lambda_e)$ be a partition of n, and f_λ be the number of SYT of shape λ. We saw that f_λ is also the number of saturated chains in $\mathbb{Y}$ from $\varnothing$ to λ.

Earlier we proved the following theorem using up and down operators:

Theorem 6.11

We have

$$\sum_{\lambda \vdash n} (f_\lambda)^2 = n!\,,$$

where the sum is over all partitions λ of n.

We will now see another proof by the *Schensted correspondence* (which was later generalized to Robinson-Schensted-Knuth, often abbreviated as RSK).

One way to think of the left-hand side combinatorially is

$$\sum_{\lambda \vdash n} (f_\lambda)^2 = \#\{(P, Q) \mid P, Q \text{ are SYT of the same shape } \lambda \vdash n\}.$$

The Schensted correspondence bijects permutations in S_n to such pairs (P, Q).

The algorithm takes as input a permutation $w = w_1 \ldots w_n \in S_n$, and outputs a pair (P, Q) of SYT of shape $\lambda \vdash n$. P is called the *insertion tableau*, and Q is called the *recording tableau*.

To illustrate how the algorithm works, we will first look at an example:

Example 6.12

Build P and Q for $w = 3524716$.

Solution. We will construct P and Q by adding boxes one at a time. Initialize P and Q as $\varnothing$.

Now we insert the first letter 3 into the first row of P. We next try to insert the second letter 5 into the first row, so that it remains increasing. Since $5 > 3$, we can insert it on the right:

$$\varnothing \implies \begin{array}{|c|}\hline 3 \\ \hline\end{array} \implies \begin{array}{|c|c|}\hline 3 & 5 \\ \hline\end{array}.$$

Now we try to insert 2 into the first row. The 2 wants to be where 3 is, but 3 is in the way: so the 2 *bumps* the 3 out of the way, and we insert 3 into the second row:

$$\begin{array}{|c|c|}\hline 2 & 5 \\ \hline 3 \\ \cline{1-1}\end{array}.$$

Now we insert 4. The 4 bumps 5 out of the way, and we insert 5 into the second row:

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

Now we can add 7 at the end of the first row:

$$\begin{array}{|c|c|c|}\hline 2 & 4 & 7 \\ \hline 3 & 5 \\ \cline{1-2}\end{array}.$$

Now when we insert 1, it bumps 2. Then when we insert 2 to the second row, it bumps 3. So we insert 3 into the third row:

$$\begin{ytableau} 1 & 4 & 7 \\ 2 & 5 \\ 3 \end{ytableau}.$$

Finally, when we insert 6, it bumps 7, which we insert into the second row:

$$\begin{ytableau} 1 & 4 & 6 \\ 2 & 5 & 7 \\ 3 \end{ytableau}.$$

So we have built the insertion tableau P. Now the recording tableau Q just keeps track of the order in which we added boxes:

$$\begin{ytableau} 1 & 2 & 5 \\ 3 & 4 & 7 \\ 6 \end{ytableau}.$$

This gives a pair of (P, Q) of the same shape $\lambda \vdash n$. □

Algorithm 6.13 (Schensted Insertion) — Start with an empty tableaux, and insert entries of w one at a time. If we currently have an intermediate tableau $\tilde{P}$, then to insert a:

- If a is larger than all entries in row 1, add a new box to the end of row 1, and place a in that box.
- Otherwise, find the smallest entry $a_1 > a$ in the first row. Replace a_1 with a.
- Repeat the same procedure to insert a_1 into the second row: if a_1 is larger than all entries in the second row, add a new box at the end. Otherwise find the smallest $a_2 > a_1$ in the second row, and replace a_2 with a_1.
- Repeat the same procedure to insert a_2 into the third row, and so on.

This constructs P; meanwhile, we label boxes of Q with the order in which they were added.

Example 6.14

Insert 4 into the first row of

$$\begin{ytableau} 1 & 2 & 5 & 7 \\ 3 & 8 \end{ytableau}.$$

Solution. The smallest entry larger than 4 is 5, so 4 bumps 5. Then 5 bumps 8, and 8 adds a box to the third row. So we end up with

$$\begin{ytableau} 1 & 2 & 4 & 7 \\ 3 & 5 \\ 8 \end{ytableau}.$$

□

Theorem 6.15

The map $w \mapsto (P, Q)$ is a bijection between S_n and the set

$$\{(P, Q) \mid P, Q \text{ are SYT of the same shape } \lambda \vdash n\}.$$

Proof. We can undo the procedure step by step, which gives an inverse map. We won't prove this in general, but we will do it for our example (which illustrates how the undoing process works). We start with

$$P = \begin{array}{|c|c|c|}\hline 1&4&6\\\hline 2&5&7\\\hline 3\\\cline{1-1}\end{array} \text{ and } Q = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4&7\\\hline 6\\\cline{1-1}\end{array}.$$

By looking at Q, we can tell that the last step added the box in P with 7. Then 7 must have been displaced by 6, so 6 must have been added into the first row. So $w_7 = 6$, and in the previous step, we had

$$P = \begin{array}{|c|c|c|}\hline 1&4&7\\\hline 2&5\\\cline{1-2} 3\\\cline{1-1}\end{array} \text{ and } Q = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4\\\cline{1-2} 6\\\cline{1-1}\end{array}.$$

Now the last box added is the box with 3. So 3 must have been displaced by 2, and 2 must have been displaced by 1. This means $w_6 = 1$, and in the previous step we had

$$P = \begin{array}{|c|c|c|}\hline 2&4&7\\\hline 3&5\\\cline{1-2}\end{array} \text{ and } Q = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline 3&4\\\cline{1-2}\end{array}.$$

Now the last box added was 7, so $w_5 = 7$. We can continue doing this process, and eventually we will recover the original permutation. □

The shape λ of P and Q is called the *Schensted shape* of w: for example, $w = 3524713$ has the Schensted shape

$$\lambda = \begin{array}{|c|c|c|}\hline &&\\\hline &&\\\hline \\\cline{1-1}\end{array}.$$

There is a lot of interesting information about a permutation contained in its Schensted shape:

Theorem 6.16

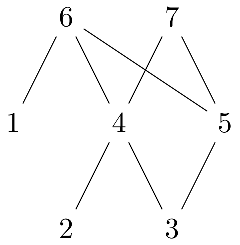
Let λ be the Schensted shape of w.

1. λ_1 (the number of squares in the first row) is the maximal size of an increasing subsequence of w.
2. λ_1' (the number of squares in the first column) is the maximal size of a decreasing subsequence.
3. Moreover, λ is the partition associated with the permutation poset of w from Greene's Theorem.

Example 6.17

Verify the theorem for $w = 3524716$.

Solution. We can draw its permutation poset as:



The maximal chain is 356, so $\ell_1 = 3$. The maximal union of chains is 356 and 247, so $\ell_2 = 6$. Then $\ell_3 = 7$. So $\lambda = (3, 3, 1)$, which is the Schensted shape we saw earlier.

Symmetrically, we can consider antichains and find that $m_1 = 3$ (from 145), $m_2 = 5$ (from 145 and 67), and $m_3 = 7$. So $\mu = (3, 2, 2)$, which is the conjugate of λ. □

Theorem 6.18

If $w \mapsto (P, Q)$ in RSK, then $w^{-1} \mapsto (Q, P)$.

We won't prove this.

§6.2.1 321-Avoiding Permutations

Now we will look at a special case of RSK. For $w \in S_n$, the following are equivalent:

1. w has no decreasing subsequence of size 3;
2. w is 321-avoiding;
3. The Schensted shape of λ is a Young diagram with at most two rows. (Equivalently, $\lambda_1' \le 2$.)

(1) and (2) are equivalent by the definition of pattern avoidance. (3) follows directly from the above theorem on Schensted shapes.

So then RSK gives a bijection between 321-avoiding permutations in S_n, and pairs of SYT (P, Q) of the same shape $\lambda \vdash n$ with at most two rows. But given such a pair (P, Q), we can combine them to get a SYT of shape n – rotate Q and replace entries $i \mapsto 2n + 1 - i$.

Example 6.19

Construct the (n, n) tableaux corresponding to $w = 312645$.

Solution. To build P, we first insert 3 to get

$$\begin{array}{|c|}\hline 3 \\ \hline\end{array}.$$

Then 1 bumps 3, giving

$$\begin{array}{|c|}\hline 1 \\ \hline 3 \\ \hline\end{array}.$$

Then we insert 2 and 6 to get

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 6 \\ \hline 3 & \multicolumn{1}{|c}{} & \\ \cline{1-1}\end{array}.$$

Now 4 bumps 6 and we get

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 & 6 \\ \cline{1-2}\end{array}.$$

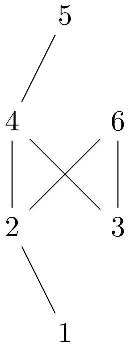
Finally, we insert 5 and get

$$P = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 4 & 5 \\ \hline 3 & 6 \\ \cline{1-2}\end{array}.$$

Then looking at insertion order,

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

The permutation poset of w is the following:



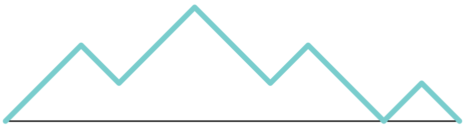
We can see that the maximal chain is 1245, and the two chains 1245 and 36 cover everything.

Now in order to construct the (n, n) SYT, we rotate Q and flip all its entries:

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 4 & 5 & 8 & 11 \\ \hline 3 & 6 & 7 & 9 & 10 & 12 \\ \hline\end{array}.$$

(We can recover P and Q from the bigger tableaux – Q consists of the entries greater than n.) □

Earlier, we saw a bijection between Young Tableaux of shape (n, n) and Dyck paths, where the ith step is a $+$ if i is in the top row, and a $-$ if i is in the bottom row. For example, our above tableaux would correspond to $++-++--+--+-$, or the following Dyck path:



This is one way of seeing that there are exactly C_n 321-avoiding permutations.

§7 Schubert Polynomials

(Guest lecture by Professor Anna Weigandt.)

§7.1 Context

Schubert polynomials are a family of polynomials defined by Lascoux and Schutzenberger in 1982, in order to study a geometric space called a *flag variety*: the set of chains of vector spaces

$$\mathrm{Fl}(n) = \{V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_n = \mathbb{C}^n\}$$

(where the inclusions are strict). These are in some sense a generalization of the Grassmannian (which we saw earlier): instead of having one subspace, we now have a chain of them. (These are called flag varieties because we can take a point, a line through that point, and a plane through that line, and this looks somewhat like a flag.)

The flag variety has a *cohomology ring* with a nice basis:

$$H^*(\mathrm{Fl}(n)) = \mathrm{Span}\{\sigma_w\},$$

where the σ_w are *Schubert classes* indexed by permutations $w \in S_n$ – these Schubert classes form a basis for the cohomology ring. (The cohomology ring tells us something geometric about how things in the space intersect; we'll treat it as a black box that combinatorialists are interested in. For example, $H^*(\mathrm{Gr}(k, n))$ is directly related to tableaux combinatorics.)

A key question is what happens when we multiply two Schubert classes: we have

$$\sigma_u \cdot \sigma_v = \sum_{w \in \sigma_n} c_{uv}^w \sigma_w.$$

There is a ring-theoretic isomorphism

$$H^*(\mathrm{Fl}(n)) \xrightarrow{\text{Borel}} \mathbb{Z}[x_1, \dots, x_n]/I$$

for some ideal I of the polynomial ring $\mathbb{Z}[x_1, \dots, x_n]$. So rather than think about Schubert classes, we can think about their images under Borel's isomorphism: consider the coset which σ_w maps to. We can take a representative of this coset $\mathfrak{S}_w$, which is a *Schubert polynomial*: so

$$\sigma_w \mapsto [\mathfrak{S}_w].$$

Now our equation becomes

$$\mathfrak{S}_u \mathfrak{S}_v = \sum_{w \in S_\infty} c_{uv}^w \mathfrak{S}_w.$$

Remark 7.1. Here we used the notation S_∞. There is an inclusion map $S_n \hookrightarrow S_{n+1}$ where we simply append $n + 1$ to the permutation. Using this inclusion map, S_∞ is the set of permutations of $\mathbb{N}$ which fix all but finitely many elements.

Remark 7.2. For the story of why these are called Schubert polynomials:

A long time ago, Schubert considered a geometric problem: suppose there are 4 fixed lines in 3-space. How many lines intersect all four fixed lines?

Schubert's solution was to squish two of the lines together until they intersect, and do the same for the other two lines. Now we have two pairs of intersecting lines. We can take the line through their intersection point. Each pair of intersecting lines also spans a plane, and the line where the two planes intersect must intersect all four lines. So this gives us 2 lines, and it's possible to show those are the only ones. Schubert said that as you move the lines back to their original positions, the number of solutions doesn't change.

Hilbert found this vague, so one of his problems was to make Schubert calculus precise – to find a rigorous foundation for counting intersections of various objects in space. The rigor came from the development of cohomology rings and intersection theory; and the counting came from combinatorial interpretations.

§7.2 Definitions

Notation 7.3. s_i is the permutation which swaps i and $i+1$.

Definition 7.4. The *divided difference operators* are defined as

$$\partial_i(f) := \frac{f - s_i \cdot f}{x_i - x_{i+1}}.$$

Here $s_i \cdot f$ means f, but with x_i and x_{i+1} swapped:

$$s_i \cdot f(x_1, \ldots, x_n) = f(x_1, \ldots, x_{i+1}, x_i, \ldots, x_n)$$

(where all other entries are in order).

Now we can define the Schubert polynomials recursively:

- Let $w_0 = (n, n-1, \ldots, 1)$ be the permutation with 0 elements in order. Then
$$\mathfrak{S}_{w_0} = x_1^{n-1} x_2^{n-2} \cdots x_{n-1}^1$$
is the *staircase monomial.*
- If $w_i > w_{i+1}$, then
$$\mathfrak{S}_{ws_i} := \partial_i(\mathfrak{S}_w).$$

Example 7.5

Calculate the Schubert polynomials for S_3.

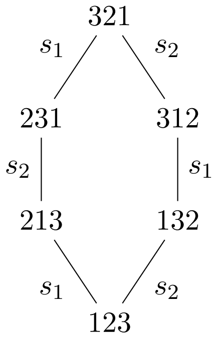
Solution. The longest permutation is 321, so $\mathfrak{S}_{321} = x_1^2 x_2$. Now we progressively find the remaining $\mathfrak{S}_w$. We have $321 \cdot s_1 = 231$, so

$$\mathfrak{S}_{231} = \partial_1(x_1^2 x_2) = \frac{x_1^2 x_2 - x_1 x_2^2}{x_1 - x_2} = x_1 x_2.$$

Similarly, $321 \cdot s_2 = 312$, so

$$\mathfrak{S}_{312} = \partial_2(x_1^2 x_2) = \frac{x_1^2 x_2 - x_1^2 x_3}{x_2 - x_3} = x_1^2.$$

(Note that we are working in $\mathbb{Z}[x_1, x_2, x_3]$, although none of our polynomials will contain an x_3 term.) We can fill in the rest of the chart similarly – the permutations have the *weak Bruhat order*:



So performing the calculations using the corresponding divided difference orders, we get:

$$\begin{tikzcd}
 & x_1^2x_2 \arrow[dl, "\partial_1"'] \arrow[dr, "\partial_2"] & \\
x_1x_2 \arrow[d, "\partial_2"'] & & x_1^2 \arrow[d, "\partial_1"] \\
x_1 \arrow[dr, "\partial_1"'] & & x_1+x_2 \arrow[dl, "\partial_2"] \\
 & 1 &
\end{tikzcd}$$

□

Proposition 7.6

The $\mathfrak{S}_w$ are well-defined: taking multiple paths to the same permutation w will give us the same result.

Proof. We know that the s_i generate S_n, under the following relations:

$$S_n = \langle s_i \mid s_i^2 = 1,\ s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1},\ s_i s_j = s_j s_i \text{ if } |i-j|> 1\rangle.$$

We can check that the latter two identities hold for ∂_i as well:

$$\partial_i \partial_j = \partial_j \partial_i$$

if $|i-j|> 1$, and

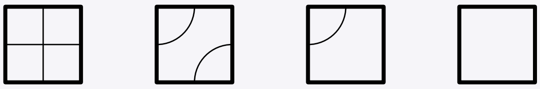
$$\partial_i \partial_{i+1} \partial_i = \partial_{i+1} \partial_i \partial_{i+1}.$$

Since we only swap if $w_i > w_{i+1}$, we don't need the first relation. (Note that ∂_i^2 is not the identity – in fact it is 0.) So any path will give us the same result. □

§7.3 Pipe Dreams

Schubert polynomials have a connection to a combinatorial object called a *pipe dream*.

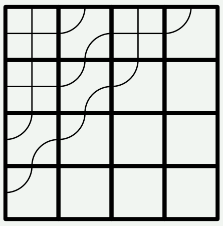
Definition 7.7. A *pipe dream* is a tiling of a $n \times n$ grid with the following four tiles:



The first and second (called *cross* and *bump* tiles) occur only above the main diagonal; the third occurs only on the main antidiagonal; and the fourth occurs only below it.

Example 7.8

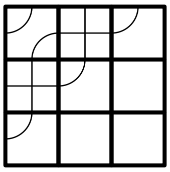
The following is a pipe dream for $n = 4$:



The name comes from the fact that the lines trace out a collection of pipes – cross tiles represent squares where two pipes cross.

Definition 7.9. A pipe dream is *reduced* if each pair of pipes crosses at most once.

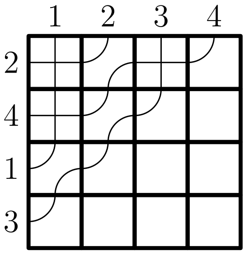
For example, the above pipe dream is reduced; meanwhile, the following is not because the two pipes on the right cross twice:



Note that a pipe dream is uniquely determined by its cross tiles.

In order to relate pipe dreams to Schubert polynomials, we want to associate each with a permutation and monomial. To find the permutation, label the top of the pipe dream with 1 through n in order, and trace out where each pipe lands on the left.

For example, the permutation associated with our first example is $w_P = 2413$:



Meanwhile, define the *weight* $\mathrm{wt}(P)$ of a pipe dream as the monomial where the exponent of x_i is the number of cross tiles in row i. For example, our above pipe dream has

$$\mathrm{wt}(P) = x_1^2 x_2^1 x_3^0 x_4^0.$$

Let Pipes(w) be the set of *reduced* pipe dreams P such that $w_P = w$.

Theorem 7.10
We have

$$\mathfrak{S}_w = \sum_{p \in \text{Pipes}(w)} \text{wt}(P).$$

This theorem was proved in many ways: by Fomin–Kirillov in 1996, Bergeron–Billey in 1993, and others. We won't prove it here, but it gives us a different way of thinking about Schubert polynomials: if we can write down Pipes(w) in a nice way, then we can use it to find $\mathfrak{S}_w$ instead of needing to do divided differences working our way down the weak Bruhat order.

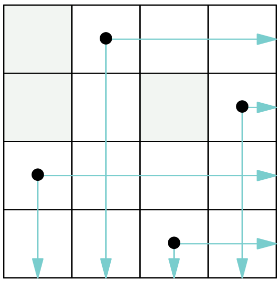
Question 7.11. How can we write down Pipes(w) for a given w?

Bergeron–Billey give an algorithmic approach using *ladder moves*.

§7.3.1 Ladder Moves

Definition 7.12. The *Rothe Diagram* of w is constructed as follows: draw a $n \times n$ grid, and plot dots at positions (i, w_i) for each i. Now draw rays down and right from each dot, and strike out all boxes in these rays. The Rothe Diagram $D(w)$ consists of the boxes which are not struck out.

For example, $w = 2413$ has the following (we use matrix notation for the pairs (i, w_i), so we read left to right and top to bottom):

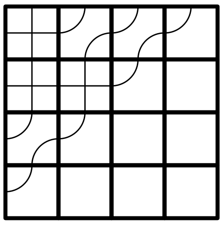


So $D(2413) = \{(1,1), (2,1), (2,3)\}$.

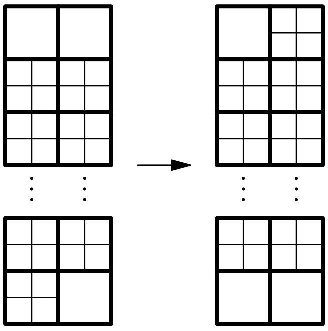
Fact 7.13 — $|D(w)| = \ell(w)$.

Here $\ell(w)$ is the *Coxeter length* of w, which equals inv(w).

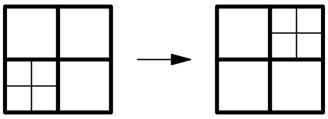
Now in order to construct Pipes(w), first take the Rothe diagram of w, turn its boxes into cross tiles, and left-justify them: so if we have k boxes in a row, then that row has cross tiles in its leftmost k squares. This gives the *bottom pipe dream* $D_{\text{bot}}(w)$. For example, when $w = 2413$, $D_{\text{bot}}(w)$ is the following:



Now given one element of $\text{Pipes}(w)$, we can perform a *ladder move* to get another. Ladder moves are local replacements on consecutive columns of our pipe dream:



Here we're only filling in the cross tiles. Essentially a cross tile jumps from the bottom-left to the top-right of a $k \times 2$ rectangle of cross tiles. The simplest case is when $k = 0$, called a *simple ladder move*:



We can see that ladder moves preserve the permutation (and the fact that the pipe dream is reduced).

Proposition 7.14

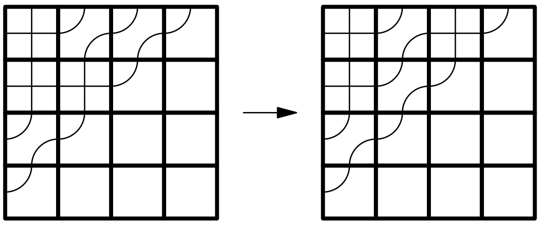
Applying ladder moves starting from $D_{\text{bot}}(w)$ produces $\text{Pipes}(w)$:

- $D_{\text{bot}}(w) \in \text{Pipes}(w)$;
- Applying ladder moves preserves w;
- All $P \in \text{Pipes}(w)$ can be reached from $D_{\text{bot}}(w)$ by a sequence of ladder moves.

Example 7.15

Find $\mathfrak{S}_{2413}$.

Solution. We found $D_{\text{bot}}(2413)$ earlier. There is exactly one ladder move which we can perform:



So these are the only two elements of $\text{Pipes}(2413)$. This means

$$\mathfrak{S}_{2413} = x_1x_2^2 + x_1^2x_2.$$

□

§7.4 Macdonald's Identity

§7.4.1 Reduced Words

We saw earlier that the simple reflections $s_i = (i, i+1)$ generate the symmetric group: every $w \in S_n$ can be written as a product of simple reflections

$$w = s_{i_1} s_{i_2} \cdots s_{i_k}.$$

Definition 7.16. If $w = s_{i_1} \cdots s_{i_k}$ where k is minimal, then $(i_1, \ldots, i_k)$ is a *reduced word* for w.

Fact 7.17 — We have $k = \ell(w)$ – the Coxeter length of w (or inversion number).

Let $\mathrm{red}(w)$ denote the set of reduced words for w.

Example 7.18
We have $\mathrm{red}(321) = \{(1,2,1), (2,1,2)\}$, since

$$321 = s_1 s_2 s_1 = s_2 s_1 s_2.$$

The different reduced words for w are connected by the *braid* and *commutation* relations we saw earlier. So if we know one reduced word for w, it is possible to describe all of them.

§7.4.2 The Identity

There is an amazing connection between reduced words and Schubert polynomials:

Theorem 7.19 (Macdonald's Identity)
We have

$$\frac{1}{\ell(w)!} \sum_{a \in \mathrm{red}(w)} a_1 \cdots a_{\ell(w)} = \mathfrak{S}_w(1, 1, \ldots, 1).$$

Note that the right-hand side counts the number of reduced pipe dreams with permutation w.

Example 7.20
Verify the identity for $w = 2413$.

Solution. We saw that $\mathfrak{S}_{2413} = x_1 x_2^2 + x_1^2 x_2$, so the right-hand side is 2. Meanwhile, we can check that

$$\mathrm{red}(2413) = \{(3,1,2), (1,3,2)\}.$$

So the left-hand side is

$$\frac{1}{3!}(3 \cdot 1 \cdot 2 + 1 \cdot 3 \cdot 2) = 2$$

as well. □

There are many proofs; we will look at a proof using derivatives.

§7.4.3 Derivatives of Schubert Polynomials

Question 7.21. How do you take the derivative of a Schubert polynomial?

Note that these are multivariate polynomials, so we'll actually take a bunch of partial derivatives. For example, we have

$$\frac{\partial}{\partial x_1}\mathfrak{S}_{2413} = 2x_1x_2 + x_2^2 \text{ and } \frac{\partial}{\partial x_2}\mathfrak{S}_{2413} = x_1^2 + 2x_1x_2.$$

In Schubert calculus, we often want to express polynomials as sums of Schubert polynomials, rather than as sums of monomials – if we do the squishy S_∞ argument mentioned earlier, then the Schubert polynomials form a basis for the ring of multivariate polynomials.

For example, we can calculate

$$\frac{\partial}{\partial x_1}\mathfrak{S}_{2413} = \mathfrak{S}_{1423} - \mathfrak{S}_{3124} + \mathfrak{S}_{2314},$$

and similarly

$$\frac{\partial}{\partial x_2}\mathfrak{S}_{2413} = \mathfrak{S}_{3124} + 2\mathfrak{S}_{2314}.$$

Minus signs are not great if we want our expressions to have combinatorial interpretations. So we can get rid of them by *adding*: instead, consider

$$\left(\frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2}\right)(\mathfrak{S}_{2413}) = \mathfrak{S}_{1423} + 3\mathfrak{S}_{2314}.$$

These polynomials are related to 2413: we can notice that 1423 is 2413 with 1 and 2 swapped – meaning we apply s_1 on the *left* to get $1423 = s_1 \cdot 2413$. Similarly, 2314 is 2413 with 3 and 4 swapped, so it's $s_3 \cdot 2413$. Meanwhile, the coefficients correspond to which s_i we multiplied by.

This is true in general:

Theorem 7.22

Let $\nabla = \sum_{i=1}^n \frac{\partial}{\partial x_i}$. Then

$$\nabla(\mathfrak{S}_w) = \sum_{s_k w < w} k\mathfrak{S}_{s_k w}.$$

This was proved by Hamaker–Pechenik–Speyer–Weigandt in 2020. We won't prove it here – the idea is to show that ∇ commutes with the divided difference operators ∂_i for each i.

Here $s_k w < w$ means that $s_k w$ has lower Coxeter length than w: so for example, our sum included $s_1 \cdot 2413$ because 1423 has fewer inversions than 2413, but not $s_2 \cdot 2413$ because 3412 has more inversions than 2413.

§7.4.4 Proof of Identity

The idea of the proof is to consider repeatedly applying ∇ to $\mathfrak{S}_w$.

Lemma 7.23

If m is a monic monomial of degree k, then $\nabla^k(m) = k!$.

Proof. This is clearly true when there's only one variable – then applying ∇ decreases the degree, and pops out a coefficient corresponding to the current degree. But the same thing happens when we have multiple variables, if we consider the sum of all the terms produced. □

Fact 7.24 — $\mathfrak{S}_w$ is a homogeneous polynomial of degree $\ell(w)$.

So then we have

$$\nabla^{\ell(w)} = \ell(w)! \cdot \mathfrak{S}_w(1, 1, \ldots, 1).$$

On the other hand, we have a rule for computing $\nabla\mathfrak{S}_w$, using the theorem

$$\nabla\mathfrak{S}_w = \sum_{s_k w < w} k\mathfrak{S}_{s_k w}.$$

So every time we hit the sum with ∇, we pull out the k from s_k and go down one step. This gives

$$\nabla^{\ell(w)}(\mathfrak{S}_w) = \sum_{a \in \mathrm{red}(w)} a_1 \cdots a_{\ell(w)}.$$

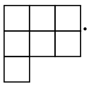
So we've computed $\nabla^{\ell(w)}(\mathfrak{S}_w)$ in two ways, and combining them gives the desired identity.

Remark 7.25. Elements in $\mathrm{red}(w)$ are in bijection with saturated chains from the identity permutation to w in the weak Bruhat order – every time we hit w with ∇, we step down to one fo the elements it covers in the poset. The paths we trace down correspond exactly to reduced words, and the weights (from the coefficients k) are exactly the products of the elements in those reduced words.

§8 Partitions

Let $p(n)$ denote the number of partitions of n – or equivalently, the number of Young diagrams with n boxes. (Young diagrams of partitions are also called *Ferrers diagrams*, where we draw dots instead of boxes – these are often more convenient to draw.)

For example,

$$\lambda = (3, 3, 1) =$$  .

We can calculate the first few values of $p(n)$:

n	$p(n)$
0	1
1	1
2	2
3	3
4	5
5	7
6	11

There is no exact, simple formula for $p(n)$. However, there is an asymptotic formula:

Theorem 8.1 (Hardy-Ramanujan)
We have

$$p(n) \sim \frac{1}{4n\sqrt{3}} e^{\pi\sqrt{2n/3}}.$$

(We will not prove this.)

§8.1 Generating Functions

Theorem 8.2 (Euler 1740)
The generating function for $p(n)$ is

$$\sum_{n\geq 0} p(n)x^n = \prod_{k\geq 1} \frac{1}{1-x^k}.$$

Proof. We can write $\lambda = (1^{m_1}, 2^{m_2}, \ldots)$, where m_i is the multiplicity of the part i in λ.

For example, $\lambda = (3, 3, 1)$ can be written as $(1^1, 2^0, 3^2, 4^0, 5^0, \ldots)$.

We then have $n = \sum_{i\geq 1} im_i$. So we can write

$$\sum_{n\geq 0} p(n)x^n = \sum_{m} x^{m_1+2m_2+3m_3+\cdots}$$

(where the sum is over all sequences $m = (m_1, m_2, \ldots)$ where the m_i are nonnegative integers, and there are finitely many nonzero m_i in each sequence). We can split this as the infinite product

$$\sum_{m_1\geq 0} x^{m_1} \cdot \sum_{m_2\geq 0} x^{2m_2} \cdot \sum_{m_3\geq 0} x^{3m_3} \cdots.$$

Each of these is an infinite geometric progression, so we get

$$\frac{1}{1-x}\cdot\frac{1}{1-x^2}\cdot\frac{1}{1-x^3}\cdots,$$

as desired. □

Remark 8.3. The product is infinite, but for any fixed n, only the first n factors contribute to x^n – so the coefficient of any monomial is finite.

We will now consider certain special partitions.

Definition 8.4. Define $p_{\text{odd}}(n)$ as the number of partitions $\lambda \vdash n$ with $\lambda = (\lambda_1, \lambda_2, \ldots)$, such that all the λ_i are odd.

Definition 8.5. Define $p_{\text{dist}}(n)$ as the number of partitions $\lambda \vdash n$ with $\lambda = (\lambda_1, \lambda_2, \ldots)$, such that all parts are distinct:

$$\lambda_1 > \lambda_2 > \lambda_3 > \cdots.$$

We can calculate the first few values of both:

n	p_{odd}	p_{dist}
0	1	1
1	1	1
2	1	1
3	2	2
4	2	2
5	3	3

Note that these are always equal.

Theorem 8.6 (Euler 1748)

For all n, $p_{\text{odd}}(n) = p_{\text{dist}}(n)$.

Proof. We can calculate the generating functions for both sequences. For partitions with odd parts, we can use the same argument as earlier: now the m_i for i even are all zero, so we only sum over m_i with i odd. So

$$\sum p_{\text{odd}}(n)x^n = \frac{1}{1-x}\cdot\frac{1}{1-x^3}\cdot\frac{1}{1-x^5}\cdots.$$

Meanwhile, for partitions into distinct parts, all the m_i are 0 or 1. So we get

$$\sum p_{\text{dist}}(n)x^n = (1+x)(1+x^2)(1+x^3)\cdots.$$

We want to show that these two infinite products are equal. We can take the first and artificially introduce more factors, to get

$$\frac{(1-x^2)(1-x^4)(1-x^6)\cdots}{(1-x)(1-x^2)(1-x^3)\cdots}.$$

Now using the fact that $(1+a)(1-a) = 1-a^2$, we can cancel corresponding terms and get that this equals

$$(1+x)(1+x^2)(1+x^3)\cdots,$$

as desired. □

Remark 8.7. It is possible to find a direct bijection between partitions with odd parts and partitions with distinct parts.

Remark 8.8. There are many nice identities of this form in the theory of partitions.

§8.2 Pentagonal Number Theorem

Consider the infinite product

$$\prod_{k\geq 1}(1-x^k) = (1-x)(1-x^2)(1-x^3)\cdots.$$

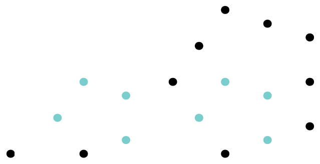
We can expand the first few terms as

$$1 - x - x^2 + x^5 + x^7 - x^{12} - x^{15} + x^{22} + x^{26} - \cdots.$$

Notice that all coefficients are either 0 or ±1, and the set of numbers which appear as powers is sparse. This is surprising – a priori you wouldn't expect either behavior of an infinite product.

Definition 8.9. The *pentagonal numbers* are the number of dots in a pentagon, constructed by adding one layer at a time. The formula for pentagonal numbers is

$$\frac{k(3k-1)}{2}.$$



The first few pentagonal numbers are 1, 5, 12, 22, 35, ... As we can see, these correspond to half of the nonzero exponents in our expansion; taking $k \leq 0$ gives the other half.

Theorem 8.10 (Euler 1750)

We have

$$\prod_{k\geq 1}(1-x^k) = \sum_{k\in\mathbb{Z}}(-1)^k x^{k(3k-1)/2}.$$

This theorem gives an efficient way to calculate the partition numbers: we have

$$\sum_{n\geq 0} p(n)x^n \cdot \prod_{k\geq 1}(1-x^k) = 1.$$

Now we know the expansion of the second term, so

$$\sum_{n\geq 0} p(n)x^n \cdot (1 - x - x^2 + x^5 + x^7 - \cdots) = 1.$$

We can expand this to get a recurrence relation:

Corollary 8.11

We have

$$p(n) = p(n-1) + p(n-2) - p(n-5) - p(n-7) + \cdots,$$

where $p(0) = 1$ and $p(n) = 0$ for $n < 0$.

This is far more efficient than listing out all possible partitions.

Remark 8.12. This theorem has connections to number theory and modular forms, and to the theory of infinite-dimensional Lie algebras.

Proof of Pentagonal Number Theorem. Consider the expansion

$$\prod_{m \geq 1} (1 - x^m) = (1-x)(1-x^2)(1-x^3)\cdots,$$

and look at the coefficient of x^n. A partition $\lambda \vdash n$ with k distinct parts contributes $(-1)^k$, so the coefficient of x^n is the sum of $(-1)^k$ over all such λ.

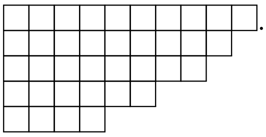
We want to show this expression is 0 for most n, and $(-1)^k$ for the pentagonal numbers $n = \frac{k(3k-1)}{2}$ for $k \in \mathbb{Z}$.

To do this, we'll use the *Involution Principle*: we want to construct an involution (a map which is its own inverse) on the set of almost all partitions with distinct parts, where one element of each pair contributes $+1$, and the other contributes -1.

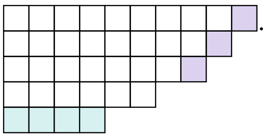
In this case, we want to send $\lambda \leftrightarrow \tilde{\lambda}$ such that $|\lambda| = |\tilde{\lambda}|$ and

$$\#\text{ parts in } \lambda = \#\text{ parts in } \tilde{\lambda} \pm 1.$$

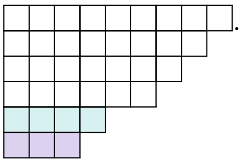
We'll do this by example: take $\lambda = (10, 9, 8, 6, 4)$, which corresponds to the Ferrers diagram



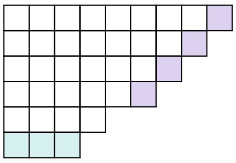
We now draw two special subsets of dots – let A be the dots on the rightmost diagonal, and B the dots in the last row:



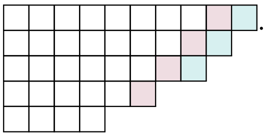
Case 1 ($|A| < |B|$). Then we take the dots in A, and move them down to the last row. For example, our above example gives:



Case 2 ($|A| \geq |B|$). Then we do the reverse: move the last row up to form a new diagonal. For example,



becomes


.

It's clear that this is an involution, and the parity fo the number of parts changes.

The exceptional cases are when A and B overlap, and $|A|=|B|$ or $|A|=|B|-1$. If A and B overlap, most of the time this is fine – if $|A|<|B|$, we can still remove the last diagonal and add it at the end; but if $|A|=|B|-1$ then we end up with two rows of the same size (because we've moved an element from the row A). Similarly, if $|A|=|B|$, when we move the last row up to a new diagonal, we end up with a diagonal that's one step too long (since we deleted the last row).

For all other partitions, the involution works – so all partitions except the ones described will cancel out. These exceptions give us exactly the pentagonal numbers – the $|A|=|B|$ case corresponds to $n=\frac{k(3k-1)}{2}$, and the $|A|=|B|-1$ case to $n=\frac{\ell(3\ell+1)}{2}$. □

§8.3 Jacobi Triple Product Identity

We've seen Euler's Pentagonal Number Theorem

$$\prod_{n\geq 1}(1-x^n)=\sum_{k\in\mathbb{Z}}(-1)^k x^{k(3k-1)/2}.$$

There are two similar formulas, proved by Gauss:

Theorem 8.13

We have the identities

$$\prod_{n\geq 1}(1-x^n)^3=\sum_{k\in\mathbb{Z}}(-1)^k k x^{k(k+1)/2}$$

and

$$\prod_{n\geq 1}\frac{1-x^n}{1+x^n}=\sum_{k\in\mathbb{Z}}(-1)^k x^{k^2}.$$

Remark 8.14. There do not exist nice equations for most other powers: for example, $\prod(1-x^n)^2$ is a mess. The reason that the first and third powers work nicely and the second doesn't is that there's a Lie algebra of dimension 3; there exists a general formula where the number is replaced by the dimension of a semisimple Lie algebra.

These three formulas are all actually special cases of a more general formula, the Jacobi triple product (which was proved by Jacobi in 1829).

Theorem 8.15 (Jacobi Triple Product Formula)

We have the identity

$$\prod_{n=1}^{\infty}(1-x^{2n})(1+x^{2n-1}y^2)(1+x^{2n-1}/y^2)=\sum_{k\in\mathbb{Z}}x^{k^2}y^{2k}.$$

Substituting values of x and y into this formula gives the identities above: for example, to get the Pentagonal Number Theorem, we can take $x=q^{3/2}$ and $y^2=-q^{1/2}$.

Proof. First, substitute $x^2 = q$ and $y^2 = zq^{1/2}$. Then we can rewrite the identity in terms of q and z, as

$$\prod_{n\geq 1}(1-q^n)(1+zq^n)(1+z^{-1}q^{n-1}) = \sum_{r\in\mathbb{Z}} z^r q^{r(r+1)/2}.$$

Divide both sides by the first term, so we want to show

$$\prod_{n\geq 1}(1+zq^n)(1+z^{-1}q^{n-1}) = \sum_{r\in\mathbb{Z}} z^r q^{r(r+1)/2}\cdot\prod_{n\geq 1}\frac{1}{1-q^n}.$$

The factor $\prod \frac{1}{1-q^n}$ is the generating function for all partitions. Meanwhile, the term $\prod(1+zq^n)$ is a generating function for the number of partitions with distinct parts: it equals

$$\sum_{\mu} z^{\#\text{ parts}} q^{|\mu|},$$

where the sum is over all μ with distinct parts. Similarly, the second term is

$$\sum_{\nu} z^{-\#\text{ parts}} q^{|\nu|-\#\text{ parts}},$$

over all ν with distinct parts.

So we want to construct a bijection between

$$\{(\mu,\nu)\mid\text{partitions with distinct parts}\}\longleftrightarrow\{(r,\lambda)\mid r\in\mathbb{Z}\text{ and }\lambda\text{ any partition}\},$$

with a few properties: we want

$$r = \#\text{ parts in }\mu - \#\text{ parts in }\nu$$

(from equating the powers of z), and

$$|\mu|+|\nu|-\#\text{ parts in }\nu = \frac{r(r+1)}{2}+|\lambda|$$

(from equating the powers of q). This would create a one-to-one correspondence between the left-hand side and right-hand side.

To describe this bijection: suppose we have μ and ν with a and b distinct parts. Then we convert them into the *shifted shapes* where we justify them on a diagonal instead of left-justifying them; call these shapes $\tilde{\mu}$ and $\tilde{\nu}$. (So $\tilde{\mu}$ has the same number of dots in each row, but the leftmost dots form a diagonal rather than a column.)

Now reflect $\tilde{\nu}$ over its diagonal to get $\tilde{\nu}'$, and glue the two diagrams together along the shared diagonal (deleting all diagonal boxes of ν).

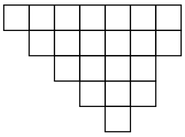
This gives us a Young diagram and a triangle; the triangle has side r, and the Young diagram is λ.

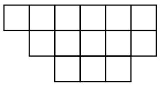
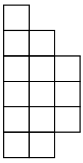
So we want to show that $r = a - b$ and $|\mu|+|\nu|-b = \frac{r(r+1)}{2}+|\lambda|$. We'll illustrate this by example.

Example 8.16

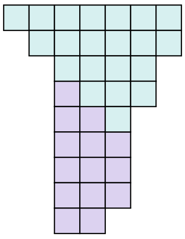
Take the partitions $\mu = (7,6,4,3,1)$ and $\nu = (6,5,3)$.

Solution. Then we have $a = 5$ and $b = 3$. We can draw the shifted shapes

$\tilde{\mu} =$  for μ, and similarly

$\tilde{\nu} =$  $\Longrightarrow \tilde{\nu}' =$  .

Now we can glue together:



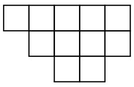
The triangle on the left has side length $r = 2$, and $\lambda = (5, 5, 4, 4, 3, 3, 3, 2)$. □

This is how the construction works when $a \geq b$. Meanwhile, we can consider an example with $a < b$ as well:

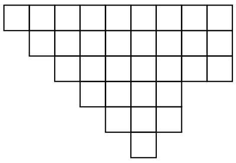
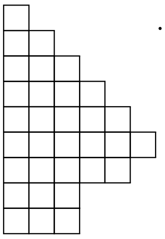
Example 8.17

Take $\mu = (5, 4, 2)$ and $\nu = (9, 8, 7, 4, 3, 1)$.

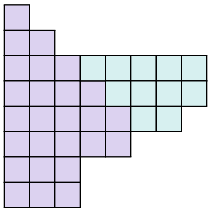
Solution. Here we have $a = 3$ and $b = 6$. Now

$\tilde{\mu} =$ 

and

$\tilde{\nu} =$  $\Longrightarrow \tilde{\nu}' =$  .

Now we glue together on the diagonal, but we also remove all the extra diagonal dots of $\tilde{\nu}'$. So we get

 .

□

This corresponds to $r = -3$ (corresponding to the triangle having side length 2 and being backwards – the formula for the number of boxes in the triangle still works here), and $\lambda = (8, 8, 7, 5, 3, 3)$. □

§8.4 q-Binomial Coefficients

Recall that

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

Many identities involving q-binomial coefficients can be proven using partitions, by finding bijections between partitions as we did earlier.

For example, in normal binomial coefficients, we have the identity

$$\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k}^2.$$

This has a q-analog:

Theorem 8.18

We have

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

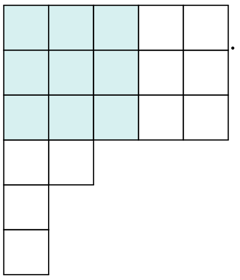
Proof. The left-hand side is

$$\sum_{\lambda \subset n \times n} q^{|\lambda|},$$

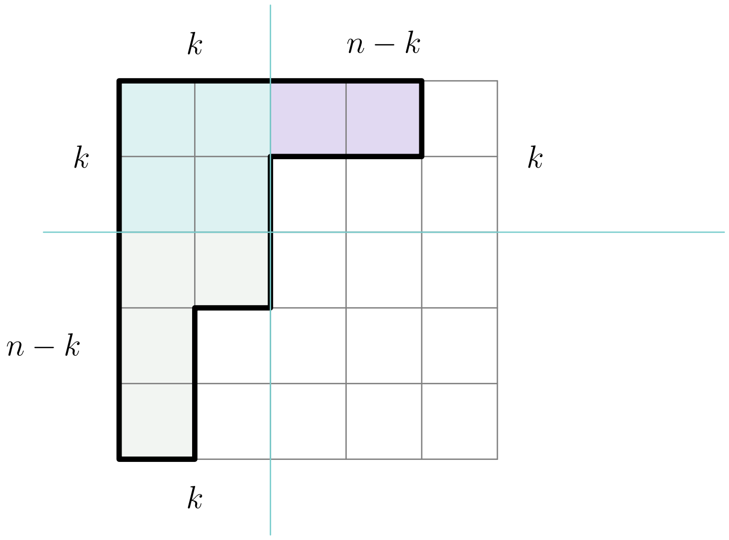
so we are fitting a Young diagram into a $n \times n$ square. Meanwhile, we can fit a square into a Young diagram as well:

> **Definition 8.19.** For a Young diagram λ, its *Durfee square* is the largest square that fits inside λ (which is axis-aligned and contains the top-left corner).

For example, the Durfee square of $(6, 5, 5, 2, 1, 1)$ has size 3:



Now if the Durfee square has size k, we can split our Young diagram in the $n \times n$ square into its Durfee square, and smaller Young diagrams $\mu \subseteq k \times (n-k)$ and $\nu \subseteq (n-k) \times k$.



It's clear that $|\lambda|= k^2 + |\mu|+|\nu|$, so we get

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

and since $\left[\begin{smallmatrix} n \\ n-k \end{smallmatrix}\right]_q = \left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right]_q$, this proves the identity. □

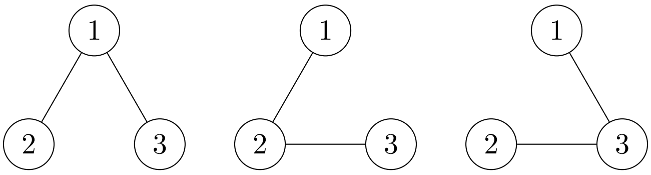
This is just one example of how various operations on Young diagrams can be used to prove identities.

§9 Spanning Trees

§9.1 Cayley's Formula

Theorem 9.1 (Cayley's Formula)
The number of labelled trees on n vertices is n^{n-2}.

We'll always assume that the labels are numbers from 1 through n. For example, the labelled trees on 3 vertices are:



Remark 9.2. The theorem is known as Cayley's Formula, but it was already known to Sylvester in 1857, and a full proof was given in a paper of Borchard in 1860. Cayley gave a proof of a generalization in 1889.

We will look at three different proofs.

§9.1.1 Algebraic Proof

This proof was given by Renyi in 1967.

We'll try to prove the formula by generating functions and induction. If we try to prove the formula directly by induction, we quickly realize it doesn't work – if we decrease n by 1, the numbers n^{n-2} and $(n-1)^{n-3}$ are not obviously related (unlike binomial coefficients, where $(n-1)! \mid n!$). So we'd end up needing to prove a complicated identity that's harder than proving Cayley's formula.

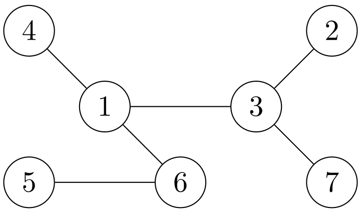
But if we can't prove something by induction, it can sometimes help to generalize it.

Definition 9.3. Define a polynomial

$$F_n(x_1, \dots, x_n) = \sum_T x_1^{\deg_T(1)-1} x_2^{\deg_T(2)-1} \cdots x_n^{\deg_T(n)-1}$$

where the sum is over all labelled trees T on vertices 1, …, n.

For example, the tree



contributes the monomial $x_1^3 x_3^2 x_6$.

Example 9.4

Calculate $F_3(x_1, x_2, x_3)$.

Solution. The trees contribute x_1, x_2, and x_3, so $F_3(x_1, x_2, x_3) = x_1 + x_2 + x_3$. □

Theorem 9.5

For all n, we have

$$F_n(x_1, \ldots, x_n) = (x_1 + \cdots + x_n)^{n-2}.$$

It's actually easier to prove this formula by induction on n – this is because if we have a stronger statement, then the inductive hypothesis is stronger as well.

Proof. Define a new polynomial

$$G_n(x_1, \ldots, x_n) = F_n(x_1, \ldots, x_n) - (x_1 + \cdots + x_n)^{n-2},$$

so we want to show G_n is identically 0. Use induction on n.

Lemma 9.6

G_n is a homgeneous polynomial in n variables, of degree $n - 2$.

Proof. Every tree contributes the monomial

$$\prod_{i=1}^{n} x_i^{\deg_T(i)-1},$$

so the degree of the monomial is

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

But the sum of degrees counts every edge twice, and a tree has $n - 1$ edges, so this is

$$2(n-1) - n = n - 2.$$

So every monomial has degree $n - 2$. □

Lemma 9.7

If we set $x_i = 0$ for any i, then

$$G_n(x_1, \ldots, x_n) = 0.$$

Proof. All vertices are symmetric, so it suffices to show that

$$F_n(x_1, \ldots, x_{n-1}, 0) = (x_1 + \cdots + x_{n-1})^{n-2}.$$

But for most trees, the monomial contributed by the tree in $F_n(x_1, \ldots, x_{n-1})$ will vanish – the only trees for which it doesn't are the trees where $\deg_T(n) = 1$, meaning n is a leaf.

Now when we remove the leaf n, we get a tree on the vertices 1 through $n-1$, which contributes to

$$F_{n-1}(x_1, \dots, x_{n-1}) = (x_1 + \cdots + x_{n-1})^{n-3}$$

by the inductive hypothesis. Meanwhile, we can add n back by connecting it to any vertex $i < n$; this increases the degree of x_i by 1. So over all possibilities, we multiply by $x_1 + \cdots + x_{n-1}$ to get

$$F_n(x_1, \dots, x_n, 0) = (x_1 + \cdots + x_{n-1})^{n-2},$$

as desired. □

So now we know G_n has degree less than n, and if $x_i = 0$ for any i, then $G_n = 0$. But this implies G_n is identically 0: if it were not identically 0, it would contain some monomial $x_1^{a_1} \cdots x_n^{a_n}$ with nonzero coefficient, and $a_1 + \cdots + a_n < n$. Then there must be some i with $a_i = 0$. When we plug in $x_i = 0$, this monomial remains nonzero; so the specialization of the polynomial to $x_i = 0$ is nonzero, contradiction. □

So this shows

$$F_n(x_1, \dots, x_n) = (x_1 + \cdots + x_n)^{n-2},$$

and setting all the x_i to 1 gives Cayley's Formula.

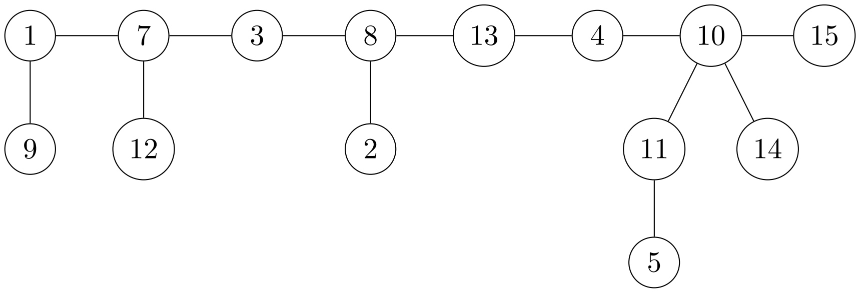
§9.1.2 Bijective Proof

We will now discuss a bijective proof found by Egecioglu and Remmel in 1986. We'll construct a bijection

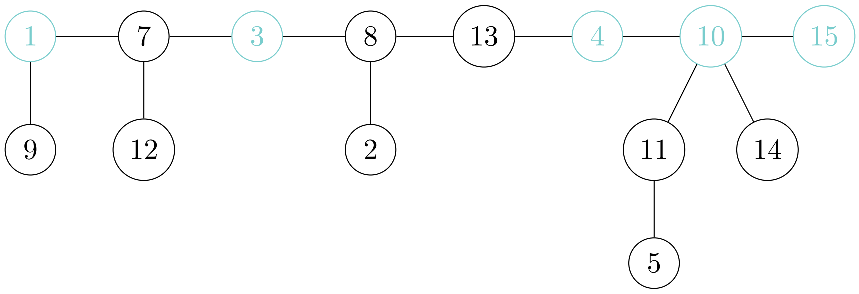
$$\{\text{trees } T \text{ on } n \text{ vertices}\} \longleftrightarrow \{\text{maps } f : [n] \to [n] \text{ s.t. } f(1) = 1 \text{ and } f(n) = n\}.$$

The second set clearly has cardinality n^{n-2} (since there are $n-2$ elements, with n choices each); so this will prove that the first set has cardinality n^{n-2} as well.

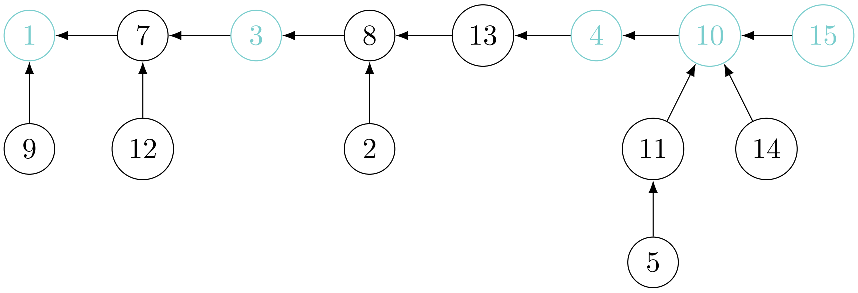
We'll illustrate the bijection by example. First draw the tree with the path from 1 to n horizontal, and the other vertices hanging down from this path.



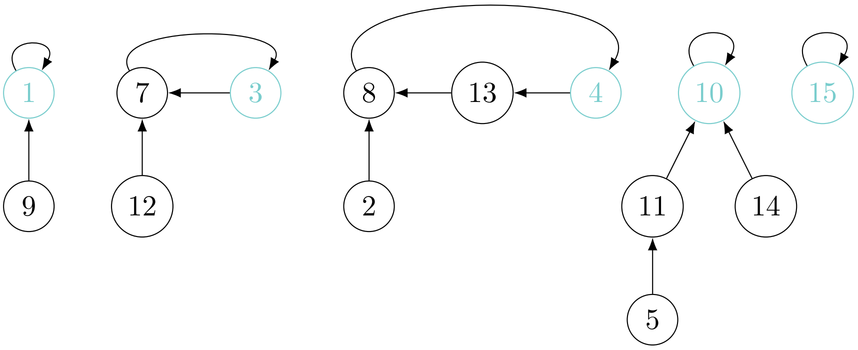
Now on the path from 1 to n, mark all vertices which are less than all vertices to their right (on the path), called *right-to-left minima.*



Now direct all the edges to point towards vertex 1.



Now, for every marked vertex, erase its incoming edge from the path. This splits the long path into a collection of short paths whose right ends are the marked vertices; now complete each path into a cycle, by drawing an edge from the left end to the marked vertex.



This creates a collection of cycles, with trees hanging off from them; this corresponds to the arrows representation of a map $f : [n] \to [n]$, where we draw arrows $x \to f(x)$.

Lemma 9.8

This map $T \mapsto f$ is a bijection.

Proof. We want to construct an inverse map $f \mapsto T$.

Every function $f : [n] \to [n]$ fixing 1 and n corresponds to a directed graph where all out-degrees are 1, and the minimal and maximal vertices are self-loops. Such a graph consists of several cycles (which may be fixed points), along with trees entering vertices of the cycle.

For each cycle, mark its minimal vertex. Now arrange the cycles on a line, such that from left to right, they are in the order of the increasing minimal vertex; and within cycles, the marked vertex is on the right.

Now replace the list of cycles with a path and undirect the edges; this recovers the tree. □

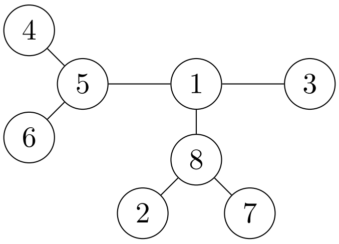
§9.1.3 Prufer Codes

Now we will look at another bijective proof, discovered by Prufer in 1918. The idea is to construct a bijection

$$T \to \text{code}(T) = (c_1, \ldots, c_{n-2}),$$

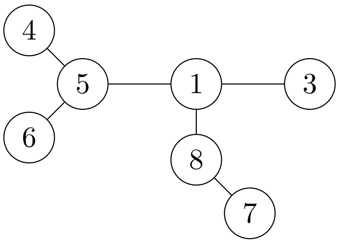
where $c_i \in [n]$ for all i.

For example, consider the following tree:

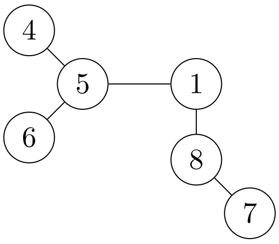


Fact 9.9 — Every tree (with at least two vertices) has at least two leaves.

Now pick the minimal leaf: here, that leaf is 2. It's attached to 8; so now delete 2, and record 8.



Now do this again: the minimal leaf is 3, and it's attached to 1; so we delete 3 and record 1.



Now the minimal leaf is 4, so we delete 4 and we record 5. Then we delete 6 and record 5. Then we delete 5 and record 1. Finally, we delete 1 and record 8. Now we're left with just a single edge between 7 and 8; once we're left with a single edge we stop. The Prufer code is the sequence we recorded: here it's

$$815518.$$

In order to show this is a bijection, we want to reconstruct the tree given the sequence. This uses one key observation:

Lemma 9.10

The number of times a vertex i appears in $\text{code}(T)$ is $\deg(i) - 1$.

Proof. We record i once for each vertex it's attached to (which gets deleted as a leaf) except the last – on the last step, we're deleting i as a leaf, and recording the vertex it's attached to instead. □

In particular, this means i is a leaf in T iff it doesn't appear in the sequence. Now we can reverse the process step-by-step. We'll illustrate this again by example:

Example 9.11

Find the tree corresponding to 815518.

Solution. Keep the structures $C = (8, 1, 5, 5, 1, 8)$ to store the code, and $L = \{1, 2, 3, \dots, 8\}$ to store the labels. We know that the leaves of T are exactly the indices not in C. Since 1 is in T and 2 isn't, we know that 2 is the minimal leaf at the first step. Since the first element of C is 8, we know that 2 was attached to 8.

Now cross out 2 from L, and cross out the first element 8 of C. Then repeat: the minimal element of L not in C is 3, and the first element of C is 1. So we know the deleted leaf was 3, and it was attached to 1. So now delete 3, and the first element 1 of C.

Now the minimal element of L not in C is 4, which is connected to 5. After deleting the 5, it's 6, which is also connected to 5. After deleting that 5, it's 5, connected to 1. After deleting 1, it's then 1, connected to 8.

Now we've created $n - 2$ edges and emptied the code, and L contains two elements 7 and 8; finally, draw an edge between 7 and 8. □

In general, we can use this same process to figure out what vertex we deleted at each step, and where it was attached; so we can recover the tree uniquely.

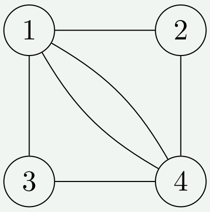
§9.2 Matrix Tree Theorem

Definition 9.12. Let G be a graph on (V, E), with $V = [n]$. A *spanning tree* of G is a subgraph $T = (V, E')$ where $E' \subseteq E$, such that T is a tree.

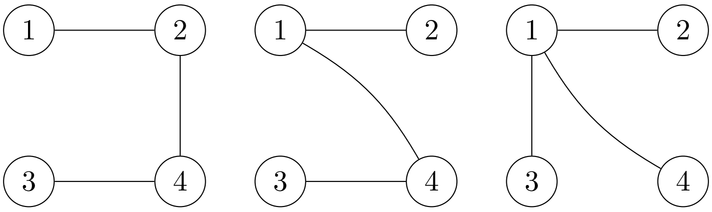
We can assume the graph has no self-loops (since these can't appear in a spanning tree).

Example 9.13

Consider the following graph:



Solution. A few of the spanning trees are:



Note that *which* of the two edges from 1 to 4 we uses does matter. This graph has 12 spanning trees. □

Cayley's Theorem can be rephrased in terms of spanning trees: it states that the number of spanning trees of K_n is n^{n-2}. We can try to find the number of spanning trees of a general graph as well.

We can encode a graph using several matrices.

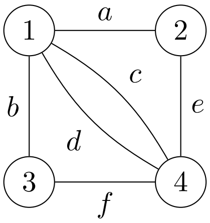
Definition 9.14. The *adjacency matrix* is the $n \times n$ matrix $A = (a_{ij})$, where a_{ij} is the number of edges between i and j.

In a simple graph, all entries of the adjacency matrix are 0 or 1. In our above graph, we have

$$A = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix}.$$

Definition 9.15. The *incidence matrix* $B = (b_{ij})$ is a $n \times |E|$ matrix whose entries b_{ij} are 1 if vertex i is incident to the jth edge, and 0 otherwise.

We say a vertex is *incident* to an edge if the edge contains the vertex. In order to create the incidence matrix, we have to label the edges. For example, if we label our graph as follows:



Then the incidence matrix is

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

Definition 9.16. The *Laplacian matrix* (also called the Kirchoff matrix) $L = (\ell_{ij})$ is the $n \times n$ matrix $L = D - A$, where A is the adjacency matrix, and D is the diagonal matrix whose diagonal entries are the degrees of the vertices.

In our example, we have

$$L = \begin{bmatrix} 4 & -1 & -1 & -2 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -2 & -1 & -1 & 4 \end{bmatrix}.$$

Note that all row sums of the Laplacian are 0. If we let the column vectors be $v_1, \ldots, v_n$, then

$$v_1 + v_2 + \cdots + v_n = 0.$$

This means the columns are linearly dependent, so $\det(L) = 0$.

Theorem 9.17 (Matrix Tree Theorem)

Fix $i \in [n]$, and define the *reduced Laplacian* $\tilde{L}$ as the $(n-1)\times(n-1)$ matrix obtained from L by removing its ith row and column. Then the number of spanning trees of G equals $\det(\tilde{L})$ for any i.

By default we usually use $i = n$. But sometimes we can choose i in order to amke the computation more convenient (since determinants are easier to calculate when they contain a lot of 0s).

Example 9.18

Use the Matrix Tree Theorem for $G = K_n$.

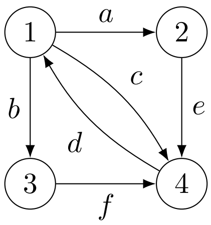
Solution. The matrix L has $n-1$ at every diagonal entry, and -1 everywhere else. The reduced Laplacian $\tilde{L}$ has the same description.

Now $\tilde{L} - nI$ is the matrix with all -1's. This matrix has rank 1. But the rank is the number of nonzero eigenvalues; so the eigenvalues of this matrix are $n-2$ 0's and one nonzero value. Since the eigenvalues sum to the trace, the last eigenvalue is $-n+1$.

So the eigenvalues of $\tilde{L}$ are $n-2$ copies of n, and one copy of 1. The determinant is the product of eigenvalues, so $\det(\tilde{L}) = n^{n-2}$. This gives another proof of Cayley's Formula. □

§9.2.1 Proof of MTT

For the easiest proof of the MTT, we'll define another matrix, the *oriented incidence matrix* $C = (c_{ij})$. First, fix any orientation of the edges in G:



Now define c_{ij} to be 1 if the jth edge starts at i, -1 if the jth edge ends at i, and 0 otherwise. For example, the above graph gives

$$C = \begin{bmatrix} 1 & 1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & -1 & -1 \end{bmatrix}.$$

Note that every column has one 1 and one -1, and 0's everywhere else.

Lemma 9.19

We have $L = C \cdot C^t$.

Note that C is a $n \times |E|$ matrix and C^T is a $|E| \times n$ matrix, so their product is a $n \times n$ matrix.

Proof. The element $(C \cdot C^t)_{ij}$ of the product is the dot product of the ith and jth rows of C.

Case 1 ($i = j$). Then we take the dot product of the row with itself. All entries are ±1, so their squares are 1; so we get the number of nonzero entries in the row, which is exactly the degree of i.

Case 2 ($i \neq j$). Then the columns where both entries are nonzero are exactly the ones corresponding to edges between i and j. Each such edge contributes $1 \cdot (-1)$, so their sum is the negative of the number of edges. □

The same reasoning also shows that $\tilde{L} = \tilde{C} \cdot \tilde{C}^t$, where $\tilde{C}$ is obtained from C by removing the ith row. So now we want to calculate $\det(\tilde{C} \cdot \tilde{C}^t)$. In order to do this, we can use a theorem:

Theorem 9.20 (Cauchy-Binet Formula)

Let B be a $k \times m$ matrix, and C a $m \times k$ matrix. Then

$$\deg(BC) = \sum_{\substack{S \subseteq [m] \\ |S| = k}} \det(B_S) \det(C_S),$$

where B_S is the $k \times k$ submatrix of B with column set S, and C_S is the $k \times k$ submatrix of C with row set S.

Here the determinants of B_S and C_S are called *maximal minors*. Note that if $m < k$, the determinant is automatically 0.

Example 9.21

Consider the matrices

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 2 & -1 & -1 \end{bmatrix} \text{ and } C = \begin{bmatrix} 3 & 5 \\ 1 & 1 \\ -2 & 3 \end{bmatrix}.$$

Solution. We can check that

$$BC = \begin{bmatrix} -1 & 16 \\ 7 & 6 \end{bmatrix},$$

which has determinant -118. Meanwhile, Cauchy-Binet gives us

$$\begin{aligned} \det(BC) &= \begin{vmatrix} 1 & 2 \\ 2 & -1 \end{vmatrix} \begin{vmatrix} 3 & 5 \\ 1 & 1 \end{vmatrix} + \begin{vmatrix} 1 & 3 \\ 2 & -1 \end{vmatrix} \begin{vmatrix} 3 & 5 \\ 2 & 3 \end{vmatrix} + \begin{vmatrix} 2 & 3 \\ -1 & -1 \end{vmatrix} \begin{vmatrix} 1 & 1 \\ -2 & 3 \end{vmatrix} \\ &= (-5) \cdot (-2) + (-7) \cdot 19 + 1 \cdot 5 \\ &= -118, \end{aligned}$$

which is the correct value. □

Proof. Consider the $(k + m) \times (k + m)$ matrices of the form

$$\left[\begin{array}{c|c} I_k & B \\ \hline 0 & I_m \end{array}\right] \cdot \left[\begin{array}{c|c} B & 0 \\ \hline -I_m & C \end{array}\right] = \left[\begin{array}{c|c} 0 & BC \\ \hline -I_m & C \end{array}\right].$$

(Here I_k and I_m are the $k \times k$ and $m \times m$ identity matrices; 0 denotes a matrix of all 0's, not necessarily square.) The determinant of the first matrix is 1, since it's upper triangular and its diagonal is all 1's. Meanwhile, the determinant of the matrix on the right-hand side is $\pm \det(BC)$, since it has the blocks BC and $-I_m$ on a diagonal. (The sign is $(-1)^{m(k+1)}$.)

Meanwhile, we claim that the determinant of the second block matrix is the right-hand side of Cauchy-Binet.

In our example, this term is

$$\left[\begin{array}{ccc|cc} 1 & 2 & 3 & 0 & 0 \\ 2 & -1 & -1 & 0 & 0 \\ \hline -1 & 0 & 0 & 3 & 5 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & -1 & -2 & 3 \end{array}\right].$$

To show this, we can use the definition of the determinant as a sum over permutations – we pick one entry from the first row, one from the second row, and so on (in different columns).

Suppose we choose 2 from the first row, and the -1 on the right in the second row. Then in order to have nonzero contribution, we need to pick the -1 in the first column of the $-I$ block. This means we can't pick 3 or 5, but we have to pick one element from each of the other two rows in C.

This argument works in general: we have to pick a $(m-k)$-element subset of the -1's, and this forces us to pick a permutation of the remaining k-element submatrices of B and C (when we delete those chosen columns or rows).

So this proves Cauchy-Binet up to signs; and if we are a bit more careful, we can see that the signs match up as well. □

Now applying Cauchy-Binet to our product, we have

$$\det(\tilde{L}) = \det(\tilde{C} \cdot \tilde{C}^t) = \sum_{\substack{S \subset E \\ |S|=n-1}} \det(\tilde{C}_S)^2.$$

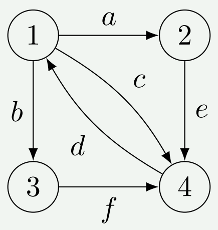
Lemma 9.22

We have

$$\det(\tilde{C}_S) = \begin{cases} \pm 1 & S \text{ forms a spanning tree} \\ 0 & \text{otherwise.} \end{cases}$$

Example 9.23

Consider $S = \{a, b, c\}$ and $\{a, d, e\}$ in our example graph:



Solution. Suppose we cross out vertex 4, so

$$\tilde{C} = \begin{bmatrix} 1 & 1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now if we take $S = \{a, b, c\}$, which does form a spanning tree, we get

$$\begin{vmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{vmatrix} = 1$$

(we can expand using the last column, which only has one 1). Meanwhile, if we take $S = \{a, d, e\}$, which doesn't form a spanning tree, we get

$$\begin{vmatrix} 1 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & 0 & 0 \end{vmatrix} = 0$$

since the last row is all 0's. □

Proof of lemma. First consider the case where S doesn't form a spanning tree. Then we have $n - 1$ edges of the graph which don't form a spanning tree; this means they contain an (undirected) cycle. If the edges of this cycle correspond to the column vectors v_{i_1}, ..., v_{i_r}, then we have

$$v_{i_1} \pm \cdots \pm v_{i_r} = 0$$

(where the signs are chosen to make the edges form a directed cycle). So a cycle implies a linear dependence among the corresponding columns.

Now consider the case where S forms a spanning tree. There must be a row with exactly one nonzero entry – such rows correspond to leaves, and every tree has at least two leaves, so at least one remains even after deleting a vertex.

Now we can expand the determinant by that row – its entry is ±1, so up to sign, this gives the submatrix where we delete that row and column. But this corresponds to deleting the leaf from the tree. So we can keep going – removing one leaf at a time – until we end up with a 1×1 matrix, giving determinant ±1. □

Remark 9.24. This property (of having determinants ±1) is known as unimodality.

§9.2.2 Eigenvalues

The area of mathematics called *spectral graph theory* tries to understand the properties of a graph G in terms of the eigenvalues of its adjacency matrix. Call the eigenvalues λ_1, ..., λ_n – these are called the *spectrum* of the graph. Many graphical properties correspond to a feature of the spectrum.

In the Matrix Tree Theorem, instead of looking at the adjacency matrix, we're looking at the Laplacian $L = D - A$. In the special case where G is d-regular (meaning all degrees are d), their eigenvalues are closely related: $L = dI - A$, so the eigenvalues of L are $d - \lambda_i$. We know one eigenvalue λ_i must be d, since L has determinant 0. In fact, this is the biggest eigenvalue; and we can find the determinant of the reduced Laplacian in terms of the others.

Lemma 9.25

Let L be any symmetric $n \times n$ matrix with 0 row sums, and eigenvalues $\lambda_1 = 0$, λ_2, ..., λ_n. Let $\tilde{L}$ be L with the ith row and column deleted. Then

$$\det(\tilde{L}) = \frac{1}{n}\lambda_2\lambda_3 \cdots \lambda_n.$$

So in order to find the number of spanning trees, it suffices to find the eigenvalues of the Laplacian. In the d-regular case, this means it suffices to find the eigenvalues of the adjacency matrix:

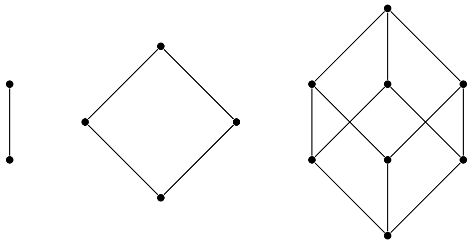
Corollary 9.26

In a d-regular graph on n vertices, if the eigenvalues of A are $\alpha_1 = d$, α_2, …, α_n, then the number of spanning trees is

$$\frac{1}{n}\prod_{i=2}^{n}(d-\alpha_i).$$

§9.2.3 Hypercubes

Let H_d be the 1-skeleton of the d-dimensional hypercube. For example, H_1, H_2, and H_3 are the following graphs:



Question 9.27. How many spanning trees does H_d have?

For example, H_1 has 1 spanning tree, and H_2 has 4 spanning trees.

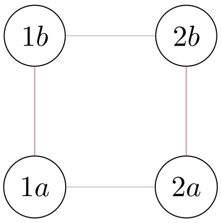
Definition 9.28. Suppose we have two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Their product $G_1 \times G_2$ is the graph (V, E) where $V = V_1 \times V_2$, and E is the set of edges (u_1, u_2) to (v_1, v_2) with $u_1, v_1 \in V_1$ and $u_2, v_2 \in V_2$, where:

- (u_1, v_1) is an edge of the first graph, and $u_2 = v_2$; or
- $u_1 = v_1$, and (u_2, v_2) is an edge of the second graph.

Example 9.29

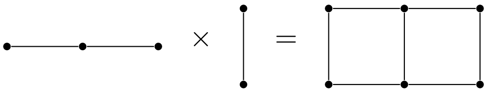
Find the product of two single edges 12 and ab.

Solution. The product is a square:

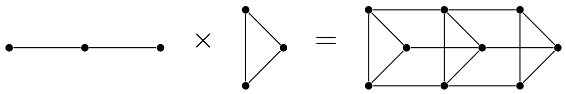


□

Similarly, the product of two paths is a grid:



The product of a path and a triangle is a triangular prism:



We then have

$$H_d = \underbrace{H_1 \times H_1 \times \cdots \times H_1}_{d}.$$

Given the spectrum of two graphs, we can find the spectrum of their product:

Lemma 9.30

Let G_1 and G_2 be two graphs, and suppose $A(G_1)$ has eigenvalues $\alpha_1, \ldots, \alpha_m$, and $A(G_2)$ has eigenvalues $\beta_1, \beta_2, \ldots, \beta_n$.

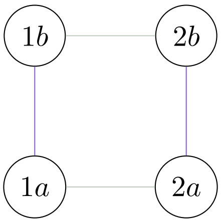
Then $A(G_1 \times G_2)$ has the mn eigenvalues $\alpha_i + \beta_j$, for $i \in [m]$ and $j \in [n]$.

This isn't hard to prove; we will look at a specific example, which should illustrate how it works in the general case as well.

Example 9.31

Consider the product of two edges 12 and ab.

Solution. We already saw this product is a square:



A single edge has adjacency matrix

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

and eigenvectors $(1,1)^t$ and $(1,-1)^t$ with eigenvalues $\alpha_1 = 1$ and $\alpha_2 = -1$, respectively. So in the ordering $1a$, $2a$, $1b$, $2b$, their product has adjacency matrix

$$\left[\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right].$$

Here the top-left and bottom-right 2×2 boxes correspond to the blue edges from G_1, while the two red squares correspond to the edges from G_2.

To obtain the eigenvectors of this matrix, we combine the eigenvectors of G_1 and G_2. Take two copies of an eigenvector v_1 of G_1, connecting the first and second indices, and the third and fourth. Similarly take two copoies of an eigenvector v_2 of G_2, connecting the first and third indices, and the second and fourth. Then multiply the corresponding entries. For example, if both eigenvectors are $(1,1)^t$, then we'd take the vector

$$\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \times \begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \begin{bmatrix}1\cdot 1\\1\cdot 1\\1\cdot 1\\1\cdot 1\end{bmatrix} = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}.$$

We then have

$$\begin{bmatrix}0 & 1 & 1 & 0\\1 & 0 & 0 & 1\\1 & 0 & 0 & 1\\0 & 1 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \begin{bmatrix}2\\2\\2\\2\end{bmatrix}.$$

Similarly, if we took the eigenvectors $(1,1)^t$ of G_1 and $(1,-1)^t$ of G_2, we'd get the vector

$$\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \times \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix} = \begin{bmatrix}1\cdot 1\\1\cdot 1\\1\cdot(-1)\\1\cdot(-1)\end{bmatrix} = \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}.$$

Similarly, taking $(1,-1)^t$ and $(1,1)^t$ gives $(1,-1,1,-1)^t$, and taking $(1,-1)^t$ and $(1,-1)^t$ gives $(1,-1,-1,1)^t$. Their eigenvalues are $1+1$, $1-1$, $-1+1$, and $-1-1$, respectively. So the spectrum is $(2,0,0,-2)$. □

Remark 9.32. This construction is related to tensor products – here, we took the tensor products of the eigenvectors.

Now we can apply this to find the spectrum of any hypercube:

Proposition 9.33

The eigenvalues of $A(H_d)$ are the 2^d numbers

$$\underbrace{\pm 1 \pm 1 \cdots \pm 1}_{d},$$

where for each $0 \le k \le d$, the eigenvalue $d-2k$ has multiplicity $\binom{d}{k}$.

Then the eigenvalues of the Laplacian are the numbers $2k$, with multiplicities $\binom{d}{k}$. So we can apply the lemma:

Corollary 9.34

The number of spanning trees of H_d is

$$\frac{1}{2^d}\prod_{k=1}^{d-1}(2k)^{\binom{d}{k}} = 2^{2^d-d-1}\prod_{k=1}^{d} k^{\binom{d}{k}}.$$

For example, the number of spanning trees of H_2 is

$$2^{2^2-2-1}\cdot 1^2\cdot 2^1 = 4.$$

Meanwhile, the number of spanning trees of H_3 is

$$2^{2^3-3-1}\cdot 1^3\cdot 2^3\cdot 3^1 = 384.$$

As we can see, this number grows very quickly, because there's a 2^{2^d} term as well as a huge product. So hypercubes have a lot of spanning trees.

> **Remark 9.35.** This formula is hard to prove combinatorially; a combinatorial proof was found a few years ago, but it is much more complicated.

There exist generalizations of the Matrix Tree Theorem as well.

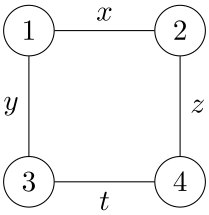
§9.2.4 Weighted MTT

Given a weighted graph $G = (V, E)$ with edge weights $x_e \in \mathbb{R}$ for every $e \in E$, we can define its weighted adjacency matrix as $A = (a_{ij})$ where

$$a_{ij} = \sum_{e=(i,j)} x_e.$$

Define the Laplacian matrix $L = (\ell_{ij})$ whose off-diagonals are $-a_{ij}$, and whose diagonal entries are the sum of weights x_e for edges e incidence to i.

For example, if we have a square:



Then the Laplacian is

$$\begin{bmatrix} x+y & -x & -y & 0 \\ -x & x+z & 0 & -z \\ -y & 0 & y+t & -t \\ 0 & -z & -t & z+t \end{bmatrix}.$$

Define T_G to be the sum over all trees of the product of edge weights:

$$T_G = \sum_T \prod_{e \in T} x_e,$$

where T ranges over all spanning trees. So in our square, we have

$$T_G = xyz + xyt + xzt + yzt.$$

> **Theorem 9.36** (Weighted MTT)
> We have $T_G = \det(\tilde{L})$.

Proof. If the weights are positive integers, then we can replace a weight-x edge with x edges. Then this follows directly from undirected MTT.

But for the general case, both sides are polynomials in the edge weights (if we fix G). If two polynomials are equal at all positive integers, they must be identically equal. □

This is not much of a generalization – it follows almost directly from the unweighted version – but it is sometimes convenient.

Example 9.37

Take G to be a triangle, with edge weights x, y, and z.

Solution. Then we have

$$T_G = xy + xz + yz.$$

Meanwhile, we have

$$L = \begin{bmatrix} x+y & -x & -y \\ -x & x+z & -z \\ -y & -z & y+z \end{bmatrix} \implies \tilde{L} = \begin{bmatrix} x+y & -x \\ -x & x+z \end{bmatrix}.$$

This means

$$\det(\tilde{L}) = (x+y)(x+z) - x^2 = xy + xz + yz$$

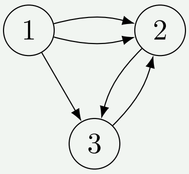
as well. □

§9.2.5 Directed MTT

Unlike the weighted version, the directed version of the Matrix Tree Theorem really is a generalization. Let G be a directed graph (also called a *digraph*).

Example 9.38

Consider the following graph:



Definition 9.39. An *arborescence* T of G with root r is a subgraph of G such that:

1. T is a spanning tree of G, when considered as an undirected graph.
2. For any vertex v, there is a directed path from the root r to v in T.

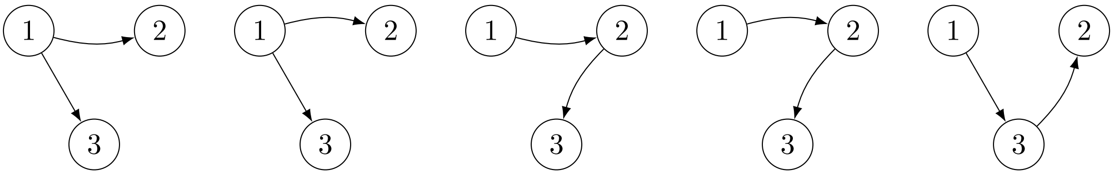
Essentially, an arborescence (or an *out-tree*) is a directed tree growing from the root.

Let $\mathrm{Arb}_r(G)$ be the number of arborescences of G with root r.

Example 9.40

Find $\mathrm{Arb}_1(G)$ for our example graph.

Solution. The arborescences are:



So $\mathrm{Arb}_1(G) = 5$. Note that $\mathrm{Arb}_2(G) = \mathrm{Arb}_3(G) = 0$, since there is no way to reach 1. □

To find the number of spanning trees, we'll define the Laplacian for directed graphs:

Definition 9.41. The directed Laplacian $L = (\ell)_{ij}$ is the $n \times n$ matrix with entries

$$\ell_{ij} = \begin{cases} -\#\{\text{edges } i \to j\} & i \neq j \\ \mathrm{indeg}(i) & i = j. \end{cases}$$

This is no longer a symmetric matrix. In our example, we have

$$L = \begin{bmatrix} 0 & -2 & -1 \\ 0 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

Note that the column sums are still 0.

Use L^{ij} to denote the *cofactors* of L – these are defined as $(-1)^{i+j}$ times the determinant of the matrix obtained by deleting the ith row and jth column.

Theorem 9.42 (Directed MTT)

For any $k, r \in [n]$, we have

$$\mathrm{Arb}_r(G) = L^{kr}.$$

Example 9.43

Verify the Directed MTT in our example graph.

Proof. In our graph, for $r = 1$ we have

$$L^{11} = (-1)^2 \begin{vmatrix} 3 & -1 \\ -1 & 2 \end{vmatrix} = 5,$$

$$L^{21} = (-1)^3 \begin{vmatrix} -2 & -1 \\ -1 & 2 \end{vmatrix} = 5,$$

$$L^{31} = (-1)^4 \begin{vmatrix} 3 & -1 \\ -1 & 2 \end{vmatrix} = 5.$$

We saw earlier that $\mathrm{Arb}_1(G) = 5$. Meanwhile, all other cofactors are 0 since we have a column of all 0's, while we saw that $\mathrm{Arb}_2(G) = \mathrm{Arb}_3(G) = 0$. □

This theorem implies the undirected version as well: given an undirected graph, replace every undirected edge uv with the two edges $u \to v$ and $v \to u$. Then L is the usual Laplacian. It's clear that if we fix the root, then spanning trees of the original graph correspond exactly to spanning trees with this root. So we actually get the more general claim that the number of spanning trees equals *any* cofactor (previously, we looked only at *principal* cofactors).

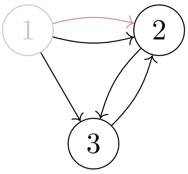
Remark 9.44. There is a dual version for in-trees as well; then we put out-degrees on the diagonal instead.

Proof. We'll use induction on the number of edges of G. The base case is when all edges of G enter the root – then assuming G has more than one vertex, there are clearly no arborescences. Meanwhile, the Laplacian has 0's everywhere except the rth column, so L^{kr} is the determinant of the 0 matrix.

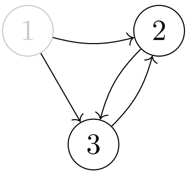
For the inductive step, pick any edge e from $i \to j$ in the graph, where $j \neq r$.

Now define G_1 to be G with the edge e deleted, and G_2 to be G with every *other* edge into j deleted – we leave e in the graph, but delete all edges $f \neq e$ which enter j.

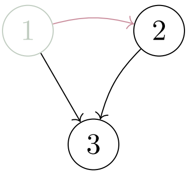
For example, suppose we choose the following edge:



Then in G_2, we delete this edge:



Meanwhile, in G_2 we remove all other edges going into vertex 2:



The key point is the following:

Lemma 9.45

We have $\mathrm{Arb}_r(G) = \mathrm{Arb}_r(G_1) + \mathrm{Arb}_r(G_2)$.

Proof. In an arborescence, there should be exactly one edge into every non-root vertex. So for an arborescence T, let f be the unique edge of T entering vertex j.

If $f \neq e$, then we can delete e, so the number of such arborescences is $\mathrm{Arb}_r(G_1)$. Meanwhile, if $f = e$, then no *other* edge in the arborescence enters j, so the number of arborescences is $\mathrm{Arb}_r(G_2)$. □

But the matrices corresponding to these graphs are closely related: in our example, these are

$$L_G = \begin{bmatrix} 0 & -2 & -1 \\ 0 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \; L_{G_1} = \begin{bmatrix} 0 & -1 & -1 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \; L_{G_2} = \begin{bmatrix} 0 & -1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}.$$

These are identical except for the jth column, while the jth column of G is the sum of the jth columns of G_1 and G_2 (this is true in general).

When finding the cofactors, we're deleting any row of the matrix, and the rth column, where $r \neq j$. So by multilinearity of determinants, we have

$$L_G^{kr} = L_{G_1}^{kr} + L_{G_2}^{kr}.$$

This nearly finishes by induction – G_1 has less edges than G. It's possible that $G = G_2$, but this only occurs when e is the only edge into j. In that case, we can contract e to get a graph G'; it's easy to check that $\mathrm{Arb}_r(G) = \mathrm{Arb}_r(G')$, and $L_G^{kr} = L_{G'}^{kr}$. So now by induction, we're done. □

> **Remark 9.46.** Note that this argument is easier than our proof of undirected MTT – proving undirected MTT by induction is much harder. We have the interesting phenomenon here that we're able to prove a stronger claim by a simpler argument – a similar thing happened when we discussed Cayley's Formula.

§9.3 Electrical Networks

An electrical network is a graph of nodes connected by wires. For each edge e, we have three numbers: the resistance $R_e > 0$, current $I_e \in \mathbb{R}$, and voltage $V_e \in \mathbb{R}$ (current and voltage can be negative). In order to discuss current, we need to direct the edges. (Reversing the direction of an edge doesn't affect R_e, but switches the sign of I_e and V_e.)

Theorem 9.47

Electrical networks satisfy the following physical laws:

1. *Kirchoff's First Law:* For every vertex v, in-current equals out-current – the sum of I_e over all e entering v is the same as the sum of I_e over all e exiting v.
2. *Kirchoff's Second Law:* For any (undirected) cycle in G with edges $e_1, \dots, e_m$, we have
$$\pm V_{e_1} \pm V_{e_2} \pm \cdots \pm V_{e_m} = 0,$$
where the signs are chosen so that the edges form a directed cycle (if an edge has incorrect direction, we negate it).
3. *Ohm's Law:* For any edge e, we have $V_e = I_e R_e$.

Lemma 9.48

Kirchoff's Second Law is equivalent to the existsence of a function U defined on the *vertices* of the graph, called the *potential*, such that for any edge $u \to v$, we have $V_e = U_v - U_u$.

These three laws then determine everything about the electrical network. Suppose we have an electrical network G, where we know all resistances. Then connect the network to a battery of known voltage V_{bat}, which connects two vertices 1 and n. Then this creates currents through the wires; our goal is to figure out all currents and voltages.

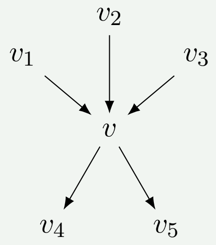
First, using Kirchoff's Second Law we can take a potential function, and write $V_e = U_v - U_u$ for every edge $u \to v$. Then Ohm's Law says that

$$I_e = \frac{V_e}{R_e} = \frac{U_v - U_u}{R_e}.$$

Now we want to write down what Kirchoff's First Law says.

Example 9.49

Find the equation from Kirchoff's First Law on v:



Solution. The first law states that

$$I_1 + I_2 + I_3 = I_4 + I_5.$$

Substituting the expressions from Ohm's Law, this means

$$\frac{U_v - U_{v_1}}{R_1} + \frac{U_v - U_{v_2}}{R_2} + \frac{U_v - U_{v_3}}{R_3} = \frac{U_{v_4} - U_v}{R_4} + \frac{U_{v_5} - U_v}{R_5}.$$

Moving everything to one side, we get

$$\sum_{i=1}^{5} \frac{U_v - U_{v_i}}{R_i} = 0.$$

Note that once we've expressed everything in terms of potential, the direction of the edges doesn't matter – the equations only depend on the undirected graph. □

Let I be the current through the battery (which is not part of the graph). Now if v is connected to v_i for $1 \le i \le d$, we have the equation

$$\left(\frac{1}{R_1} + \cdots + \frac{1}{R_d}\right) U_v - \left(\frac{U_{v_1}}{R_1} + \cdots + \frac{U_{v_d}}{R_d}\right) = \begin{cases} 0 & v \neq 1, n \\ I & v = 1 \\ -I & v = n. \end{cases}$$

This is a system of linear equations, so we can write it down in a compact form using matrices. *Kirchoff's Matrix* is the $n \times n$ matrix (k_{uv}) where k_{uu} is $\sum \frac{1}{R_e}$ over all e incident to u, while for $u \neq v$, k_{uv} is $-\sum \frac{1}{R_e}$ over all edges e connecting u and v. Note that this is exactly the Laplacian matrix for graph G with edge weights $\frac{1}{R_e}$.

Now if we define $\vec{U} = (U_1, \dots, U_n)^t$ as the vector of potentials, then the system of equations becomes

$$K\vec{U} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \\ -I \end{bmatrix}.$$

There is a problem: recall that the determinant of the Laplacian is 0, so this system doesn't have a unique solution. But this is clearly true, since we can increase all potentials by a constant. So we can fix this easily by requiring $U_n = 0$.

Then we will get a system of $n-1$ equations with $n-1$ variables – let $\tilde{K}$ be the *reduced* Kirchoff matrix, where we delete the nth row and column. Now this is equivalent to

$$\tilde{K} \cdot \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{n-1} \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

We also don't know I. But we do know that U_1 is the voltage of the battery. This is fine, because of scaling – we can solve the equations assuming $I = 1$, and then rescale to make U_1 the correct value.

Now we have a complete way to figure out all voltages and currents.

Suppose that the vertices connected to the battery are a and b. Then the entire graph is equivalent to a single resistor between a and b, whose resistance is called the *effective resistance* $R_{ab}(G)$.

We can actually find an explicit expression for the effective resistance. By Ohm's Law, we know

$$R_{ab}(G) = \frac{V_{\text{bat}}}{I_{\text{bat}}} = \frac{U_1}{I}.$$

To keep things simple, assume $I = 1$. Then we want to solve for U_1 in

$$\tilde{K} \cdot \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{n-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

But by Kramer's Rule, we can get

$$U_1 = \frac{\det \tilde{\tilde{K}}}{\det \tilde{K}},$$

where $\tilde{\tilde{K}}$ is $\tilde{K}$ with the first row and column deleted. (This is because in order to find the determinant on the top, we take $\tilde{K}$ and replace the first column by the vector $(1, 0, \ldots, 0)^t$ – this is equivalent to the determinant of the bottom-right $(n-2) \times (n-2)$ submatrix.)

But the determinant of the Laplacian is the number of spanning trees of the graph. So then electrical networks are actually closely related to spanning trees:

Theorem 9.50

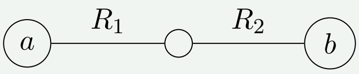
Let G be the graph corresponding to the electrical network with edge weights $\frac{1}{R_e}$, and G' the graph obtained from G by gluing together a and b. Then we have

$$R_{ab}(G) = \frac{\det K^{ab}}{\det K^{b}} = \frac{\sum_{T'} \text{wt}(T')}{\sum_{T} \text{wt}(T)},$$

where T' ranges over all spanning trees of G', and T over all spanning trees of G (and $\text{wt}(T)$ denotes the *weight* of T, the product of all its edge weights). Here K^b is K with the bth row and column removed, and K^{ab} is K with the ath and bth rows and columns removed.

Example 9.51

Find the effective resistance of a series connection:



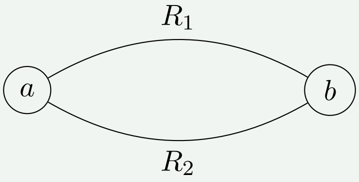
Solution. There is only one spanning tree of G, with weight $\frac{1}{R_1 R_2}$. There are two spanning trees of G' (which becomes a 2-cycle), with weights $\frac{1}{R_1}$ and $\frac{1}{R_2}$. So the effective resistance is

$$\frac{\frac{1}{R_1} + \frac{1}{R_2}}{\frac{1}{R_1 R_2}} = R_1 + R_2.$$

So in a series connection, the resistances add. □

Example 9.52

Find the effective resistance of a parallel connection:

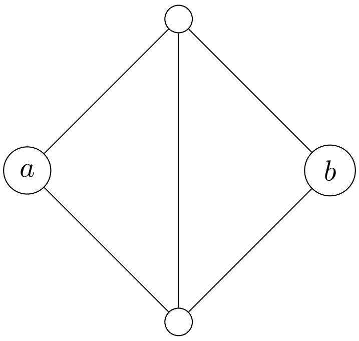


Solution. There are two spanning trees of G, with weights $\frac{1}{R_1}$ and $\frac{1}{R_2}$. Meanwhile, G' is a single vertex with two self-loops, so its only spanning tree is tree with no edges. So the effective resistance is

$$\frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}.$$

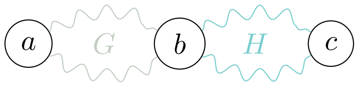
So in a parallel connection, the *reciprocals* of resistance add. □

We can use these two cases to simplify any series-parallel graph – this process is actually equivalent to computing the number of spanning trees. But there exist graphs which aren't series-parallel:



But when a graph is not series-parallel, we can still use the general theorem, by summing over spanning trees – in this case, there are 8 spanning trees in both the numerator and denominator.

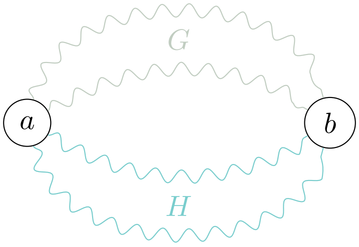
There is another way to think of series and parallel connections: the numerator $w(T')$ is a sum over *almost* spanning trees of G, meaning forests consisting of two trees where a and b are in different trees. We can get the laws for series and parallel connections from here. If we have the graphs G and H connected in series at b:



Then to find a spanning tree of the new graph, we'd take a spanning tree of G and H. Meanwhile, to get an *almost* spanning tree of the new graph, we'd take a spanning tree of one graph, and an almost spanning tree of the other. This gives

$$R_{ac}(\text{series}) = R_{ab}(G) + R_{bc}(H).$$

Similarly, if we have G and H connected in parallel, from a to b:



Then to get a spanning tree of the new graph, we'd take a spanning tree of one of G and H, and an almost spanning tree of the other; while to get an almost spanning tree, we'd take an almost spanning tree of both graphs. This gives

$$\frac{1}{R_{ab}(\text{parallel})} = \frac{1}{R_{ab}(G)} + \frac{1}{R_{ab}(H)}.$$

§9.3.1 Inverse Boundary Problem

We've seen that if we have a graph with two vertices a and b connected to a battery, we can replace the entire graph with a single resistance R_{ab}. From the point of view of measurements between the two vertices, these two graphs are indistinguishable.

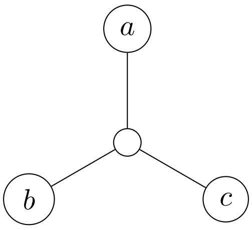
But now imagine that instead of having just two vertices where we can connect batteries, we have multiple such vertices.

> **Question 9.53.** Suppose we have a black box containing an electrical network, with wires sticking out at the *boundary vertices* $b_1, \ldots, b_n$. We don't know the graph or resistances inside the box, but we can take *boundary measurements*, where we measure the effective resistance between any pair of boundary vertices. Knowing these boundary measurements, can we reconstruct the graph?

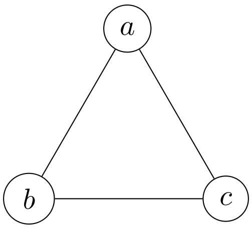
The general answer is no – a large graph with only two boundary vertices is equivalent to a single resistor. So we can't reconstruct the whole graph, but we can still try to extract some information – for example, given two graphs, we may want to figure out whether they are equivalent.

> **Question 9.54.** Are there transformations we can perform on a graph which preserve the boundary measurements?

The answer is yes: we can perform Y-Δ transforms. Suppose we have three resistors connected in an upside-down Y-shape, with resistances R_1, R_2, and R_3.



Then we can remove these three resistors, and replace them with three other resistors forming a triangle, with resistances R'_1, R'_2, and R'_3.



Proposition 9.55

There is a unique transformation $(R_1, R_2, R_3) \mapsto (R'_1, R'_2, R'_3)$ such that the two graphs have the same boundary measurements; and this transformation is invertible.

Proof. It's enough to consider the network with boundary vertices abc. Both graphs are series-parallel, so we can calculate effective resistances fairly easily: in the first graph, we have

$$R_{ab} = R_1 + R_2,$$

while in the second, we have

$$\frac{1}{R_{ab}} = \frac{1}{R'_1} + \frac{1}{R'_2 + R'_3}.$$

This gives a system of equations with 3 equations and 3 unknowns, which we can solve. □

Of course, there are also the simpler series and parallel transformations: we can replace two resistors connected in series with a single resistor of resistance $R_1 + R_2$, or two resistors connected in parallel with a single resistor of resistance $(R_1^{-1} + R_2^{-1})^{-1}$.

Theorem 9.56

If G_1 and G_2 are two *planar* electrical networks, where all boundary vertices lie on its perimeter, then G_1 and G_2 are equivalent iff they can be obtained from each other by a sequence of Y-Δ transforms, and series and parallel replacements.

This theorem was proved by Curtis, Ingerman, and Morrow in 1998, in the paper "Circular planar graphs and resistor networks."

Note that series and parallel replacements are *reduction moves* – if we think of the resistances as unknowns, then they reduce the number of unknowns. But Y-Δ transforms don't change the number of unknowns. We can call a graph *reduced* if it has the minimal number of edges among all equivalent graphs; then if two graphs are reduced, we only need to use Y-Δ transforms.

§9.4 Random Walks on Graphs

Example 9.57

A drunk person is walking on a cliff; he starts at a fixed point, and on every turn, he moves one step right with probability $\frac{1}{2}$, and one step left with probability $\frac{1}{2}$. If he reaches point 0, he falls off the cliff; meanwhile, if he reaches point n, he reaches a house and goes to sleep. What is the probability that he reaches the house?

Solution. We have $p_0 = 0$ and $p_n = 1$, and

$$p_i = \frac{1}{2}p_{i-1} + \frac{1}{2}p_{i+1}$$

for all $1 \le i \le n-1$. This is a system of $n+1$ unknowns and $n+1$ equations.

We can see that the equations imply the p_i form an arithmetic sequence, so they should be evenly spaced; so $p_i = \frac{i}{n}$ for all i. □

We can generalize this to any graph G (in this case, G was a chain), and add edge weights.

Let G be a graph with edge weights c_e (which we can think of as $\frac{1}{R_e}$), and two special vertices a and b, where a is the house and b is the cliff.

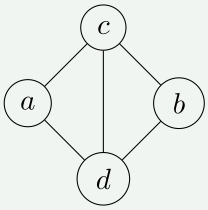
Then we randomly walk on this graph: if we are at a vertex v, then we walk to one of its neighbors v_1, ..., v_d, with probabilities proportional to the edge weights: if the weight of edge (v, v_i) is c_i, then

$$p(v, v_i) = \frac{c_i}{c_1 + \cdots + c_d}.$$

Let p_v be the probability that if we start at v, we reach a before b.

Example 9.58

Consider the following graph, where all edge weights are 1:



Solution. From a and b, there is a $\dfrac{1}{2}$ probability of going to each neighbor; from c and d there is a $\frac{1}{3}$ probability of going to each neighbor.

Here we have $p_a = 1$ and $p_b = 0$; by symmetry, we have $p_c = p_d = \frac{1}{2}$. (We could theoretically go back and forth infinitely without ever reaching a or b, but if the graph is finite, this happens with probability 0.) □

Remark 9.59. We can also think of this as a Markov chain with four states, and transitional probabilities $\frac{1}{2}$ from a and b to each neighbor, and $\frac{1}{3}$ from c and d to each neighbor.

For an arbitrary graph, we then get a system of equations: if v has neighbors v_1, ..., v_d, then we have

$$p_v = \sum_{i=1}^{d} \frac{c_i}{c_1 + \cdots + c_d} p_{v_i},$$

while $p_a = 1$ and $p_b = 0$.

So we have a system of linear equations. If we look carefully at this system, we can realize it's actually the same equation as the one we wrote for electrical networks: if we clear denominators, we get

$$(c_1 + \cdots + c_d)p_v - \sum_{i=1}^{d} c_i p_{v_i} = 0.$$

So if $c_e = \frac{1}{R_e}$, then this is the exact same equation as the one given by the Kirchoff matrix, and we have

$$K \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ -I \end{bmatrix},$$

with the conditions $p_a = 1$ and $p_b = 0$. So the probabilities are the potentials of vertices: if we connect a and b to a 1-volt battery source (so the potential of b is 0 and the potential of a is 1), then p_v is the potential of v in the electrical network.

In particular, the original drunken walk problem is equivalent to connecting a bunch of resistors in series.

§10 Parking Functions

In Cayley's Formula, we saw the expression $(n+1)^{n-1}$. It turns out this appears in many other places.

Suppose we have n cars, and n parking spots on a one-way road (where the cars and parking spots are both labelled 1, . . . , n). The cars will arrive one by one from the left (in order), and try to park.

Each car has a favorite spot – we have a *preference function* $f : [n] \to [n]$, which we can write as $(f_1, \ldots, f_n)$, such that car i has favorite spot f_i.

The cars park according to the following rule: they go to their favorite spot, and if it's not yet occupied, then they park there. Otherwise, they keep driving from there and take the next available spot. (The road is one-way, so cars cannot go backwards.)

Definition 10.1. The preference function f is called a *parking function* if all cars can park.

Example 10.2

Find all parking functions for $n = 3$.

Solution. There are three cars, and three spots. So the parking functions are the following:

- $(1, 2, 3)$ is a parking function: since there are no conflicts, every car can park in their preferred spot. Similarly, all 6 permutations are parking functions.
- $(1, 1, 3)$ is a parking function – the first car parks in spot 1, the second keeps driving and takes spot 2, and the third in spot 3. Similarly, all 3 permutations work.
- $(1, 2, 2)$ is a parking function – the first car parks in spot 1, the second in spot 2, and the third keeps driving and parks in spot 3. Similarly, all 3 permutations work.
- $(1, 1, 2)$ is a parking function – the first car parks in spot 1, the second in 2, and the third in 3. All 3 permutations work.
- Finally, $(1, 1, 1)$ is a parking function.

Meanwhile, $(1, 3, 3)$ is not a parking function – two cars will try to park in the last spot, so one will get stuck. Similarly, $(2, 2, 2)$ is not a parking function – the first car takes spot 2, the second takes spot 3, and the third gets stuck. We can check no other functions work, so the total is

$$6 + 3 + 3 + 3 + 1 = 16.$$

□

Example 10.3

For $n = 4$, determine whether $(3, 1, 3, 1)$ and $(3, 2, 3, 2)$ are parking functions.

Solution. For $(3, 1, 3, 1)$, the first car takes spot 3, and the second car takes spot 1. Then the third car goes to spot 3 and sees it's occupied, so they take spot 4; similarly, the fourth car goes to spot 1 and sees it's occupied, so they take spot 2. So this is a parking function.

Meanwhile, $(3, 2, 3, 2)$ is not a parking function because all cars drive past spot 1. □

From these examples, we can formulate a lemma describing parking functions:

Lemma 10.4

Let $f = (f_1, \dots, f_n)$ be a preference function. Then the following are equivalent:

- f is a parking function.
- f has at most 1 entry equal to n, at most 2 entries at least $n-1$, and so on – for all $1 \le k \le n$, we have
$$\#\{i \mid f_i \ge n-k+1\} \le k.$$
- There exists a permutation $w = w_1 \dots w_n$ of $[n]$ such that $f_i \le w_i$ for all i.

For example, in the case of $n = 3$, all our parking functions could be obtained by taking permutations of $(1, 2, 3)$ and decreasing some of the entries.

Theorem 10.5

The number of parking functions of size n is $(n+1)^{n-1}$.

Proof. We'll start by slightly modifying the setup. Instead of having n spots on a one-way road, we now have $n+1$ spots on a *circular* road.

The rules for parking are the same: given a preference function $f : [n] \to [n+1]$, every car i first drives to its favorite spot f_i, and tries to park; if it's taken, they keep driving and take the first available spot.

Now in this setup, the cars will always be able to park, and one spot will always be left empty. Let F_i be the set of preference functions $f : [n] \to [n+1]$ such that spot i is left empty.

Then we clearly have
$$|F_1|+|F_2|+ \dots + |F_{n+1}|= (n+1)^n,$$
since there are $(n+1)^n$ functions $f : [n] \to [n+1]$.

But all spots are symmetric, so we have $|F_1|= |F_2|= \dots = |F_{n+1}|$. The explicit bijection between F_i and F_{i+1} is to add 1 to each f_i (taken mod $n+1$).

Example 10.6

For $n = 3$, consider the parking function $f = (2, 3, 3)$.

Solution. We have four spots 1, 2, 3, and 4. The first car parks at spot 2, the second at spot 3, and the third at spot 4. So then the remaining spot is 1, and f is in F_1.

If we add 1 to all entries to get $\tilde{f} = (3, 4, 4)$, then the first car parks in spot 3, the second in spot 4, and the third in spot 1 – this is the same process, moved counterclockwise by one step. □

Finally, a preference function f in this new setup is a parking function in the original problem iff $f \in F_{n+1}$ – this means no one likes spot $n+1$, so all f_i are in $[n]$; and no car drives past spot $n+1$, so they were essentially driving on the one-way road.

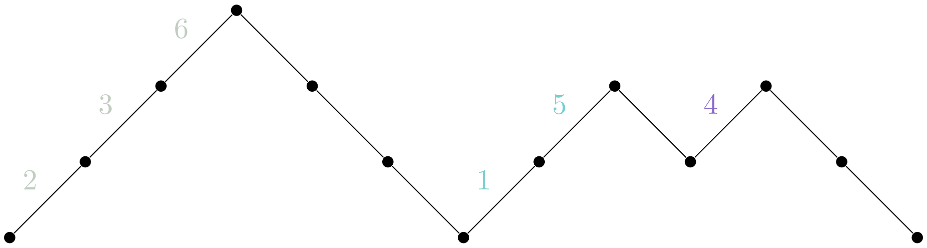
So the number of parking functions is
$$|F_{n+1}|= \frac{1}{n+1} \cdot (n+1)^n = (n+1)^{n-1},$$
as desired. □

§10.1 Bijection to Trees

We know that $(n+1)^{n-1}$ is also the number of labelled trees on $n+1$ vertices. So we can try to find a bijection between parking functions and trees.

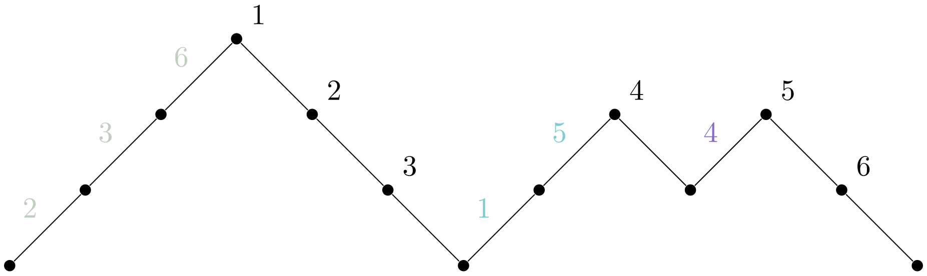
In order to do this, we'll construct an intermediate object:

Definition 10.7. A *labelled Dyck path* is a Dyck path where the up-steps are labelled from 1 to n (without repetition), such that the labels increase inside every diagonal.



For example, this is a labelled Dyck path because $2 < 3 < 6$ and $1 < 5$.

Label the diagonals 1, 2, 3, . . . , n (where some diagonals may be empty):



Proposition 10.8

There is a bijection between labelled Dyck paths with $2n$ steps, and parking functions of size n: given a labelled Dyck path P, define the bijection $P \mapsto (f_1, \dots f_n)$, where $f_i = j$ iff the up-step labelled i belongs to the jth diagonal.

Example 10.9

Find the parking function corresponding to the above path.

Solution. The first diagonal contains the labels 2, 3, and 6, so we set $f_2 = f_3 = f_6 = 1$. The second and third diagonals are empty. The fourth diagonal contains labels 1 and 5, so we set $f_1 = f_5 = 4$. The fifth diagonal contains 4, so we set $f_4 = 5$. So we get $f = (4, 1, 1, 5, 4, 1)$. □

Now we'll show this is a bijection:

Proof. First, the path stays weakly above the x-axis iff it has at least k up-steps in the first k diagonals (since after each diagonal, we have exactly one down-step). Converting to parking functions, this means there are at least k cars with $f_i \leq k$. This is equivalent to the condition given in the earlier lemma for f to be a parking function. □

Notice that if we permute the entries of a parking function, we get another parking function – this is clear from the conditions given in the lemma. But permuting the elements of a parking function corresponds to permuting labels in the Dyck path: so if we instead consider parking functions up to permutations, this gives a bijection to usual (unlabelled) Dyck paths.

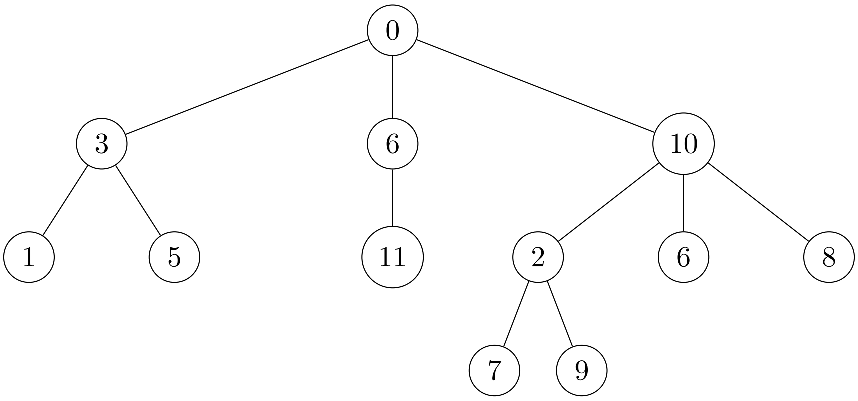
Theorem 10.10

The number of parking functions f such that $f_1 \leq \cdots \leq f_n$ is C_n.

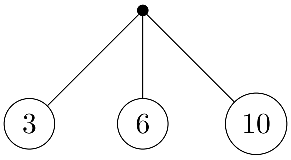
So the sum we saw in the example, $6 + 3 + 3 + 3 + 1 = 16$, will have C_n terms in general, giving a way to split $(n+1)^{n-1}$ as a sum of C_n terms.

So this gives a bijection between labelled Dyck paths and parking functions; now we want a bijection between trees and labelled Dyck paths.

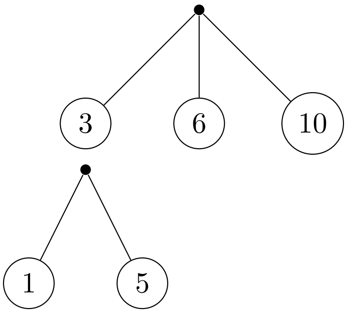
Take a labelled tree on the vertex set $\{0, 1, \ldots, n\}$, and draw T in the plane such that 0 is its root, and the children of each vertex increase from left to right:



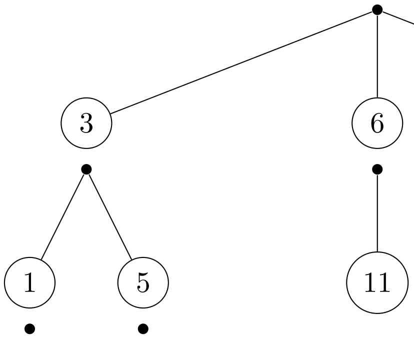
Now we break the tree into pieces: we start with a dot, with children 3, 6, and 10.



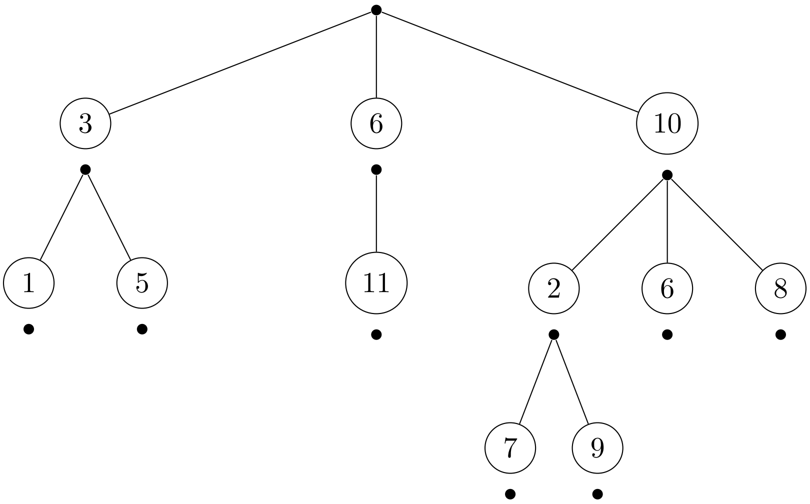
Now the leftmost place where we can attach vertices is 3, which has children 1 and 5; so we attach a dot with children 1 and 5:



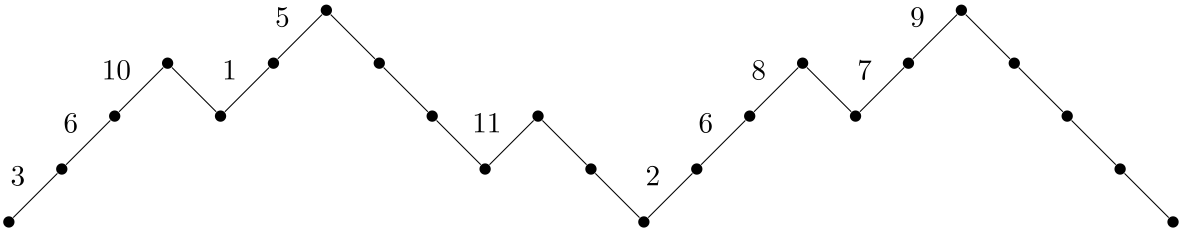
Now the leftmost place where we can attach vertices is 1; it has no children, so we attach a dot with no children there. Now the leftmost place is 5, so we attach a dot with no children at 5 as well. Then the leftmost place is 6, so we attach a dot with child 11.



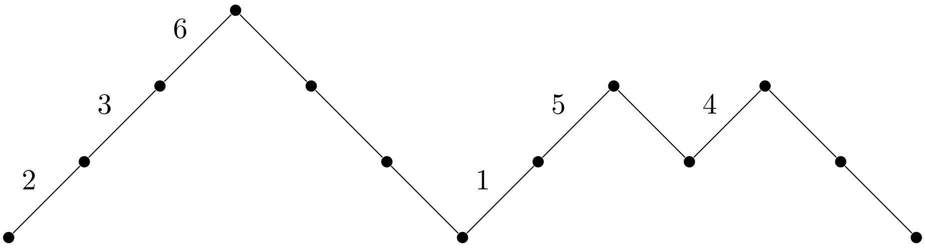
Now the leftmost place where we can attach is 11, so we attach a dot at 11. Then we attach a dot at 10, with children 2, 6, and 8; we continue until we have built the entire tree.



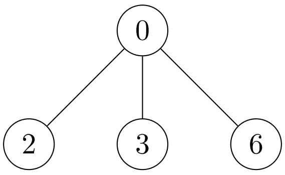
Now we turn these blocks into segments of the Dyck path. The first block we added contained 3, 6, and 10, so we write down three up-steps labelled 3, 6, and 10, followed by a down-step. The next block contained 1 and 5, so we write down up-steps labelled 1 and 5, followed by a down-step. The next block was empty, so we just write a down-step. And so on (ignoring the final dot):



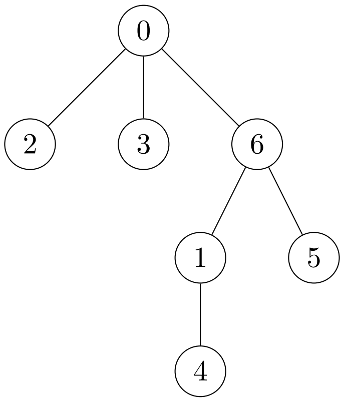
We can recover the tree from the Dyck path as well: consider our original example.



We can cut the path into segments $(2, 3, 6)$, $()$, $()$, $(1, 5)$, (4), and $()$. So then we start by attaching the children 2, 3, and 6 to 0.



The next vertex which is processed is 2; since the next segment is empty, we don't attach any children. Then we process 3 by adding no children; 6 by attaching 1 and 5; and 1 by attaching 4. (We can pretend that there is one extra down-edge, corresponding to finishing the last vertex.)



So this gives a bijection between labelled Dyck paths and labelled trees.

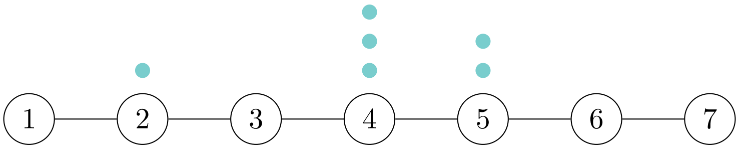
§10.2 Chip Firing

We saw that parking functions of size n correspond to spanning trees on K_{n+1}. But we can look at the spanning trees of any graph G; so we can try to generalize parking functions to an arbitrary graph as well.

This generalization is the *chip firing game*, also called the *abelian sandpile model.*

> **Remark 10.11.** The process was first introduced in a paper by the physicists Bak, Tang, and Wisenfeld in 1987, who called it the abelian sandpile model; it was independently discovered by Bjoren, Lovasz, and Shor in 1991, who called it the chip-firing game.
>
> The reason physicists were studying this process was in order to create a simple mathematical model for complicated natural processes, in particular avalanches. There are places where a simple mathematical set of rules may produce complicated and unpredictable behavior, for example fractals; this model will have similar properties.

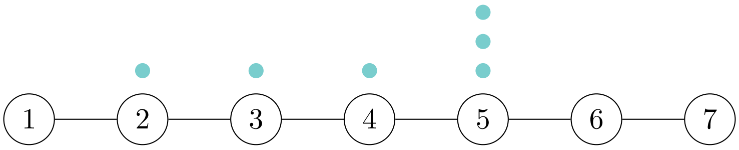
Take a graph $G = (V, E)$. At every vertex of the graph, there is some number of chips. The *configuration* c is a vector storing the number of chips at each vertex. For example, if G is a chain on 7 vertices:



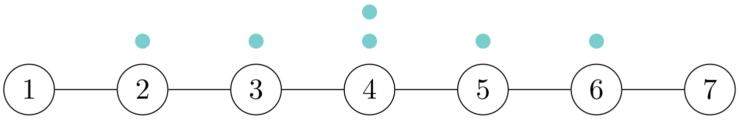
Here we have $c = (0, 1, 0, 3, 2, 0, 0)$. (The physicists thought of these chips as grains of sand or snowflakes at a given time, and the graph as a grid.)

If a vertex has at least as many chips as its degree, then it can *fire* – we move one chip from that vertex to each neighbor. (In the physical interpretation, this corresponds to the vertex toppling.)

For example, vertex 4 has 3 chips and degree 2, so it can fire. We move one chip to vertex 3 and one to 5, to get the configuration $(0, 1, 1, 1, 3, 0, 0)$:



Now we can fire at vertex 5, to get $(0, 1, 1, 2, 1, 1 0)$:



Now vertex 4 can fire again, and so on.

We keep firing until we arrive at a *stable* configuration, where no firing is possible – so $c_i < \deg(i)$ for all vertices i. (At any step, if multiple vertices can fire, then we can fire at any one of them; but we only fire one vertex at a time.)

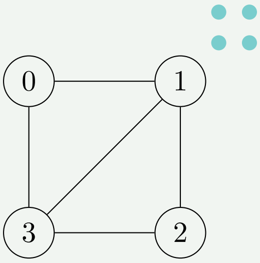
> **Remark 10.12.** This is a model for an avalanche: the position may be mostly unstable, with a few unstable vertices. Then those unstable vertices can fire and create more unstable vertices – so even if the graph was slightly unstable to begin with, we might now have a lot of unstable vertices.

There is a problem: if there are a lot of chips, then *any* configuration is unstable, and the process will go on forever. In order to fix this, we'll assume that our graph has one special vertex, called the *sink* – all chips which go into the sink disappear. (We can think of the sink as a black hole.)

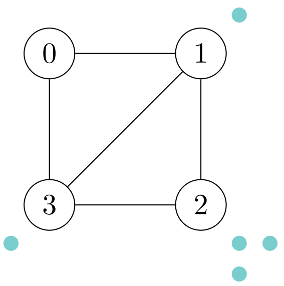
We'll assume the graph is connected, and has vertex set $\{0, 1, \ldots, n\}$, where 0 is the sink. We write the configuration vector as $c = (c_1, \ldots, c_n)$.

Example 10.13

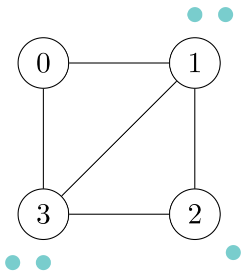
Consider the following graph, with initial configuration $(4, 2, 0)$:



Solution. We can first fire at 1, so we move one chip to 2, one chip to 3, and one chip to the sink. This gives the configuration $(1, 3, 1)$.



Now we can fire at 2, so we move one chip to 1, and one chip to 3, to get the configuration $(2, 1, 2)$.



This is stable, so we're done.

But there are other possible scenarios: at the beginning, 2 was also unstable, so we could have started there and gotten $(5, 0, 1)$. Then we'd fire at 1 and get $(2, 1, 2)$. So we end up with the same configuration. □

Lemma 10.14

For any initial configuration c_{init}, we get a stable configuration c_{stab} after finitely many firings; and the resulting configuration c_{stab} is unique.

This final configuration is called the *stabilization* of the initial configuration.

Proof. To prove finiteness, we can show that the chips move closer to the sink: assign each vertex a cost, so that vertices closer to the sink are more expensive. (For example, all chips which enter the sink cost a million dollars; all chips next to the sink cost a hundred, and so on.) It's possible to define this cost explicitly, and show that it increases every turn.

For uniqueness, the idea is that we can rearrange firings by the *diamond argument*: suppose we start with a configuration c, and we can either fire i or j, giving configurations c' and c'' respectively.

$$\begin{tikzcd} & c \arrow[dl, "i"'] \arrow[dr, "j"] & \\ c' & & c'' \end{tikzcd}$$

Use induction; so we can assume that the uniqueness claim has already been proved for c' and c''.

Now the key idea is that if i and j were both unstable and we fired i, then j is *still* unstable. So from c' we can fire j, while from c'' we can fire i. We can see that the firing process commutes, so these both give the same result.

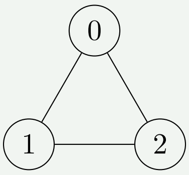
$$\begin{tikzcd}
 & c \arrow[dl, "i"'] \arrow[dr, "j"] & \\
c' \arrow[dr, "j"'] & & c'' \arrow[dl, "i"] \\
 & c''' &
\end{tikzcd}$$

But we know by induction that the uniqueness claim is true for c' and c'' (any way of firing them should lead to the same configuration); one of these ways is $c' \to c''' \to \cdots$ and $c'' \to c''' \to \cdots$, so this finishes. □

Chip-firing is related to the Laplacian: if the rows of the *reduced* Laplacian (where we cross out the row and column corresponding to the sink) are $\ell_1, \ldots, \ell_n$, then firing at i corresponds to $c \mapsto c - \ell_i$.

Example 10.15

Consider the graph K_3, with initial configuration $(3, 1)$.



Solution. Here the Laplacian is

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix},$$

and the reduced Laplacian is

$$\tilde{L} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

If we fire at 1, then we get

$$(3, 1) \mapsto (3, 1) - (2, -1) = (1, 2).$$

In order to stabilize, we then fire at 2 to get

$$(1, 2) \mapsto (1, 2) - (-1, 2) = (2, 0).$$

Finally, we fire at 1 again to get

$$(2, 0) \mapsto (2, 0) - (2, -1) = (0, 1).$$

So then $(0, 1)$ is the stabilization of the original configuration $(3, 1)$. □

§10.2.1 Abelian Sandpile Model

Definition 10.16. The *avalanche operators* $A_1, \ldots, A_n$ act on the set of stable configurations, where A_i maps c to the stabilization of $c + e_i$.

Here e_i denotes the vector with a 1 in the ith position, and 0's everywhere else. So A_i first adds a 1 to the ith position of c, and then stabilizes it.

Lemma 10.17 (Dhar)

The avalanche operators are well-defined and commute, meaning that $A_iA_j = A_jA_i$ for all i and j.

Proof. We won't go through the details, but it's not hard to see from what we've said so far. We can show that adding a chip at the ith vertex and stabilizing, and then adding a chip at the jth vertex and stabilizing, gives the same result as adding a chip at the ith *and* jth vertex, and then stabilizing. □

Remark 10.18. This property is why this is called the *abelian* sandpile model.

Now we can define the model:

Definition 10.19. The *abelian sandpile model* is the random walk (or Markov chain) on the set of stable configurations, given by the following process: randomly pick a vertex $1 \le i \le n$ (with uniform probability), and apply the avalanche operator $c \mapsto A_i(c)$.

Example 10.20

Consider the graph K_3 again.

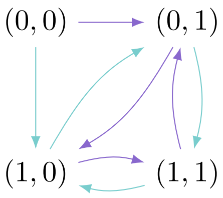
Solution. The stable configurations are $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$.

From $(0,0)$, the avalanche operator A_1 sends $(0,0) \mapsto (1,0)$, and A_2 sends $(0,0) \mapsto (0,1)$.

From $(1,0)$, the avalanche operator A_1 sends $(1,0) \mapsto (2,0) \mapsto (0,1)$, while A_2 sends $(1,0) \mapsto (1,1)$. The case $(0,1)$ works similarly.

Finally, from $(1,1)$, A_1 sends $(1,1) \mapsto (2,1) \mapsto (0,2) \mapsto (1,0)$, while A_2 similarly produces $(0,1)$.

So we get the following process, where all edges have probability $\frac{1}{2}$:



Here the blue edges represent the operator A_1, and the red edges represent A_2. As we can see, once we leave $(0,0)$ we can't reach it anymore; but we can reach all the other positions. □

Definition 10.21. A stable configuration c is *recurrent* if there exists a positive integer N such that $A_i^N(c) = c$ for all indices i.

In the above example, A_1 and A_2 both form 3-cycles, so we can take $N = 3$.

Let R denote the set of recurrent configurations.

It turns out that the abelian sandpile model is actually related to spanning trees:

Theorem 10.22

We have that $|R|$ is the number of spanning trees of G.

Recall that the number of spanning trees is $\det \tilde{L}$. In fact, there is a stronger theorem that explains this relationship better.

Lemma 10.23

The restrictions of the A_i to R are invertible.

In our above example, A_1 is not invertible because we can't come back to $(0, 0)$; but if we restrict it to the three recurrent configurations, then it is.

Proof. This is clear: we can find N such that $A_i^N(c) = c$ for all c, so then the inverse of A_i is A_i^{N-1}. □

This means we can talk about the group generated by the A_i:

Definition 10.24. The *sandpile group* SG of a graph is the group generated by $A_1, \ldots, A_n$ restricted to the set R.

Each avalanche operator is some huge permutation of R, so the sandpile group is a subgroup of $S_{|R|}$. The operators commute, so it's a finite abelian group.

Theorem 10.25 (Dhar)

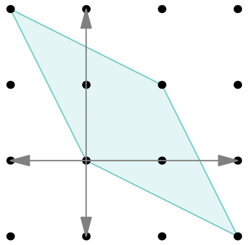
The sandpile group is isomorphic to $\mathbb{Z}^n/\langle \ell_1, \ldots, \ell_n\rangle$ (where we're quotienting by the span of the rows of the reduced Laplacian). We have

$$|R|= |SG|= \det \tilde{L}.$$

Example 10.26

Consider this theorem for the graph K_3.

Solution. We want to show that the sandpile group is isomorphic to $\mathbb{Z}^2/\langle(2, -1), (-1, 2)\rangle$. We can see that $\langle(2, -1), (-1, 2)\rangle$ is a lattice; so when we quotient $\mathbb{Z}^2$ by this lattice, we can move every point into the fundamental parallelogram (in general, the fundamental parallelepiped) generated by these two vectors.



Here the parallelogram has three points. So $\mathbb{Z}^2/\langle(2, -1), (-1, 2)\rangle$ is generated by the two elements e_1 and e_2, with relations $e_1e_2 = e_2e_1$, and $e_1^2e_2^{-1} = e_1^{-1}e_2^2 = 1$ (corresponding to quotienting by $(2, -1)$ and $(1, -2)$, respectively); these are the same relations satisfied by A_1 and A_2. □

Remark 10.27. The fact that the order of $\mathbb{Z}^n/\langle \ell_1, \dots \ell_n \rangle$ is $\det \tilde{L}$ is true in general – if we quotient by the span of n linearly independent integer vectors, then the number of elements in the quotient equals the determinant of the matrix whose rows are these vectors.

Question 10.28. Given a configuration, how can we tell whether it's recurrent?

There is a characterization of recurrent configurations:

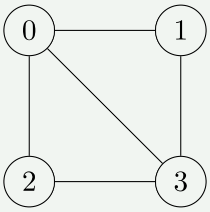
Theorem 10.29

A stable configuration $(c_1, \dots, c_n)$ is recurrent iff for any nonempty subset $I \subseteq [n]$, there exists $i \in I$ such that $c_i \geq \deg_{G_I}(i)$, where G_I is the induced subgraph on I.

This means for any subset I of the vertices, it has some vertex i such that if we count the edges from i which remain inside I, we have at least as many chips as edges.

Example 10.30

Find all recurrent configurations for the following graph:



Solution. First, a configuration (c_1, c_2, c_3) is stable iff $0 \leq c_1, c_2 \leq 1$ and $0 \leq c_3 \leq 2$, so there are 12 stable configurations.

Meanwhile, the conditions for (c_1, c_2, c_3) to be recurrent are:

- For I with one vertex, or $I = \{1, 2\}$, there are no edges in the induced subgraph; so the condition is trivial.
- For $I = \{1, 3\}$, we must have $c_1 \geq 1$ or $c_3 \geq 1$.
- For $I = \{2, 3\}$, we must have $c_2 \geq 1$ or $c_3 \geq 1$.
- For $I = \{1, 2, 3\}$, we must have $c_1 \geq 1$, $c_2 \geq 1$, or $c_3 \geq 2$.

So then the recurrent configurations are $(1, 1, 0)$, $(1, 0, 1)$, $(0, 1, 1)$, $(0, 0, 2)$, and all configurations obtained by adding chips to these which are still stable: $(1, 1, 1)$, $(1, 0, 2)$, $(0, 1, 2)$, and $(1, 1, 2)$.

So there are 8 recurrent configurations, out of the 12 stable configurations. Earlier, we calculated that this graph has 8 spanning trees. □

As mentioned earlier, the set of recurrent configurations in the complete graph can be thought of as parking functions.

Remark 10.31. We mentioned a few results, but this model is really very mysterious – if you take a large grid graph and run the chip-firing game for a long time, you'll end up with something kind of like fractals: with some random parts, and some parts with a lot of structure.

§10.3 Tree Inversions

Now we'll return to usual parking functions. Recall that the number of parking functions $(f_1, \ldots, f_n)$ is $(n+1)^{n-1}$. But we may want to study other statistics on parking functions. In particular, we'll try to count parking functions by their sum of entries $f_1 + \cdots + f_n$.

This turns out to be related to the tree inversion polynomial: let T be a labelled tree on vertices 0, 1, ..., n, where 0 is its root.

Definition 10.32. A pair (i, j) of entries with $1 \le i < j \le n$ is an *inversion* of T if j belongs to the shortest path between i and the root.

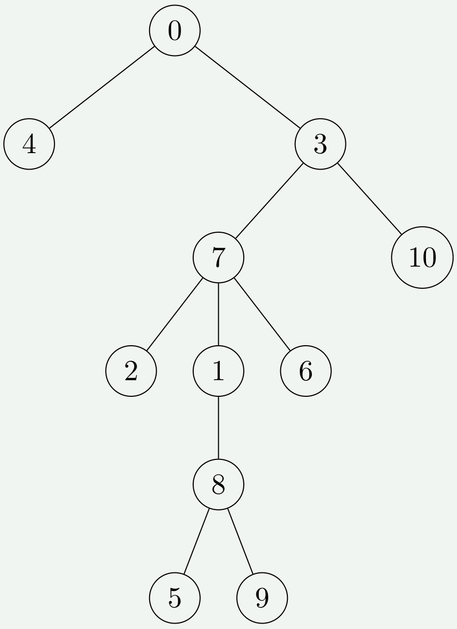
So we can draw T with vertex 0 on the top; and for a vertex i, we can draw the path upwards from i to 0. If j is on this path, with $i < j$, then (i, j) is an inversion.

If the tree is a chain, then tree inversions are inversions of the usual permutations.

Let $\operatorname{inv}(T)$ denote the number of inversions of T.

Example 10.33

Find the inversions of the following tree:



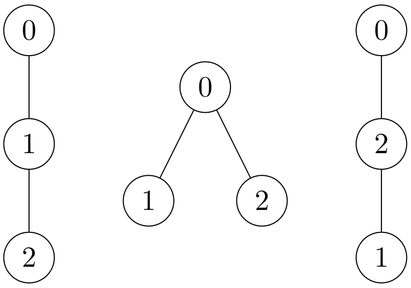
Solution. We can look at each vertex's path to the root. The inversions are $(1,7)$, $(1,3)$, $(2,7)$, $(2,3)$, $(5,8)$, $(5,7)$, and $(6,7)$. So this tree has 7 inversions. □

Definition 10.34. The tree inversion polynomial is the polynomial

$$I_n(x) = \sum_T x^{\operatorname{inv}(T)},$$

where the sum is over all trees T on $n+1$ vertices.

For example, when $n = 2$ there are three trees on 3 vertices:



The first two have no inversions, and the third has one inversion, so $I_2(x) = 2 + x$. Similarly, we can count

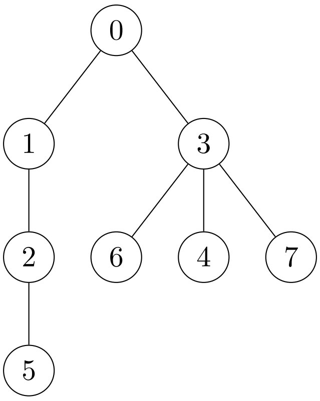
$$I_3(x) = 6 + 6x + 3x^2 + x^3.$$

There are a few observations we can make about the coefficients of I_n. First, the leading term corresponds to a tree with the maximal number of inversions – which is a chain, with the vertices in decreasing order. So the leading coefficient is always 1, and the exponent of x is $\binom{n}{2}$.

Meanwhile, the constant term corresponds to trees without inversions. So as we go downwards from the root, the labels always increase.

Definition 10.35. A tree T is called a *increasing tree* if $\text{inv}(T) = 0$.

For example:



We have $I_0 = I_1 = 1$, so we can see that the constant terms are 1, 1, 2, 6. This suggests a pattern:

Lemma 10.36

There are $n!$ increasing trees on $n + 1$ vertices.

Proof. We can prove this by induction – the idea is to attach vertices one at a time. We know that n must be a leaf, so we can attach it to any of the n previous vertices, giving n ways to attach it.

Alternatively, this also follows from the directed Matrix Tree Theorem: increasing trees are exactly arborescences of the graph with vertices 0, 1, . . . , n, where we draw directed edges $i \to j$ for all $i < j$ (so K_{n+1} with all edges directed from the smaller to larger vertex).

The number of arborescences is the determinant of the reduced Laplacian. Then $\text{indeg}(i) = i$, so in the reduced Laplacian where we remove the row and column corresponding to 0, we have 1, 2, . . . , n on the diagonal, -1's above the diagonal, and 0's below. This is an upper triangular matrix, so its determinant is the product of its diagonal entries, which is $n!$. □

Tree inversions are related to parking functions:

Theorem 10.37 (Kreweras 1980)

We have

$$I_n(x) = \sum_f x^{\binom{n+1}{2} - (f_1 + \cdots + f_n)},$$

where the sum is over all parking functions $f = (f_1, \ldots, f_n)$.

Recall that parking functions are exactly the sequences obtained by taking a permutation of $\{1, 2, \ldots, n\}$ and decreasing some entries. So the exponent measures how far f is from the maximal parking function.

We can consider special values of x. When $x = 1$, then $I_n(1) = (n+1)^{n-1}$ is the number of trees on $n+1$ vertices; this is also the number of parking functions. When $x = 0$, then $I_n(0) = n!$ is the number of increasing trees; meanwhile, parking functions with $f_1 + \cdots + f_n = \binom{n+1}{2}$ are exactly permutations, so their count is $n!$ as well.

§10.3.1 Alternating Permutations

Other values of x give interesting results as well: consider $I_n(-1)$, which counts the *alternating* sum of parking functions (where we add the functions with maximal sum, subtract the ones whose sum is one less, and so on). This turns out to be related to alternating permutations.

Definition 10.38. A permutation $w \in S_n$ is *alternating* if $w_1 < w_2 > w_3 < w_4 > \cdots$.

Let A_n denote the number of alternating permutations of size n.

Remark 10.39. These permutations have many names: the permutations are also called up-down permutations and zigzag permutations, and the A_n are also called Andre numbers, Euler numbers, zigzag numbers, up-down numbers, tangent and secant numbers, and so on. They are also related to the Bernoulli numbers.

We can calculate the first few values:

n	A_n
0	1
1	1
2	1
3	2
4	5
5	16
6	61

Theorem 10.40

We have $I_n(-1) = A_n$ for all n.

Before we prove this, we'll first look at a way to calculate all these numbers in a triangle, similar to Pascal's Triangle.

The *Euler-Bernoulli Triangle* is the following:

$$\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 0 & & 1 & & & & & \\
 & & & & 1 & & 1 & & 0 & & & & \\
 & & & 0 & & 1 & & 2 & & 2 & & & \\
 & & 5 & & 5 & & 4 & & 2 & & 0 & & \\
 & 0 & & 5 & & 10 & & 14 & & 16 & & 16 & \\
61 & & 61 & & 56 & & 46 & & 32 & & 16 & & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}$$

We alternate between filling in rows left to right and right to left; and we add the elements of the above row one at a time. For example, to construct the second-last row we write down 0, 5, $5+5$, $5+5+4$, and so on.

Then the numbers on the sides are the A_n – the left side, called the Euler or secant side, contains $A_0 = 1$, $A_2 = 1$, $A_4 = 5$, $A_6 = 61$, and so on; while the right side, called the Bernoulli or tangent side, contains $A_1 = 1$, $A_3 = 2$, $A_5 = 16$, and so on.

Remark 10.41. The numbers on the left give the coefficients of the exponential generating function for secant, and the numbers on the right give the coefficients for tangent. The person to introduce this way of writing the numbers in a triangle was Arnold, but he called it the Euler-Bernoulli triangle.

Now we'll prove the theorem $I_n(-1) = A_n$. Kreweras proved it bijectively, but the bijection is complicated (the bijection we created earlier doesn't work here, since the sum of entries doesn't correspond to inversions).

Instead, we'll see an inductive proof, which is not easy but manageable. The idea is to show that $I_n(-1)$ and A_n satisfy the same recurrence relations.

Proposition 10.42

The tree inversion polynomial satisfies the recurrence relation

$$I_n(x) = \sum_{k=1}^{n} \binom{n-1}{k-1}(1 + x + x^2 + \cdots + x^{k-1}) I_{k-1}(x) I_{n-k}(x),$$

with initial condition $I_0(x) = 1$.

Then we want to see that A_n and $I_n(-1)$ satisfy the same recurrence, so we can substitute $x = -1$. Then $1 + x + \cdots + x^{k-1}$ evaluated at $x = -1$ is 1 if k is odd, and 0 if k is even. So we want to show the following recurrence:

Proposition 10.43

The numbers of alternating permutations satisfy the recurrence

$$A_n = \sum_{k \text{ odd}} \binom{n-1}{k-1} A_{k-1} A_{n-k},$$

with initial condition $A_0 = 1$.

Remark 10.44. This recurrence looks somewhat similar to the recurrence for Catalan numbers (apart from the parity condition and binomial coefficient). This is not a coincidence – Catalan numbers correspond to unlabelled Dyck paths, and we've seen that trees correspond to labelled Dyck paths.

Proof of Proposition 10.43. Consider an alternating permutation $w_1 < w_2 > \cdots$. We want to split it into two smaller permutations; we can do this by looking at 1. If $w_k = 1$, then k must be odd (because we need w_k to be smaller than its neighbors, which occurs exactly at odd indices).

Now let the first part of the permutation be $w' = w_1 \cdots w_{k-1}$, and the second part be $w'' = w_{k+1} \cdots w_n$. Then we can count the number of ways to pick w' and w''. First, we need to choose $k-1$ entries for w', which can be any $k-1$ numbers between 2 and n; so there's $\binom{n-1}{k-1}$ ways to choose the subset $\{w_1, \dots, w_{k-1}\}$. Then there's A_{k-1} ways to arrange these elements to form w', and A_{n-k} ways to arrange the remaining ones to form w''. So this gives $\binom{n-1}{k-1} A_{k-1} A_{n-k}$ ways for each k. □

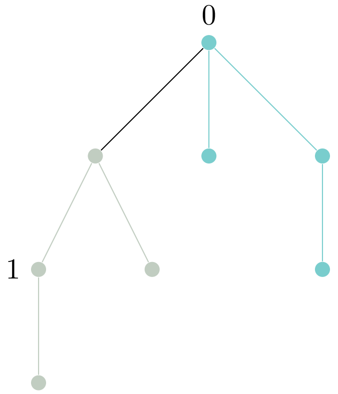
> **Remark 10.45.** We can get the same recurrence using *even* k instead, by considering n instead of 1.

Now we'll prove the recurrence for the tree inversion polynomial:

Proof of Proposition 10.42. In the first class, we found the recurrence for the Catalan numbers by splitting a Dyck path into two smaller Dyck paths. We'll use a similar idea here – we want to split a tree into two smaller trees.

Consider a tree T, with a root 0 and one or more branches coming out of the root. One of these branches contains the vertex 1.

Take T_1 to be the branch containing 1, and take T_2 to be the rest of the tree (meaning 0 together with all other branches).



Then the tree consists of T_1, T_2, and the edge from 0 to the root of T_1.

Suppose that T_1 contains k vertices (with $1 \le k \le n$), and T_2 contains $n-k+1$ vertices. Note that T_2 is rooted at its minimal vertex (which is 0), but T_1 is *not* necessarily rooted at its minimal vertex (which is 1) – let the root of T_1 be r.

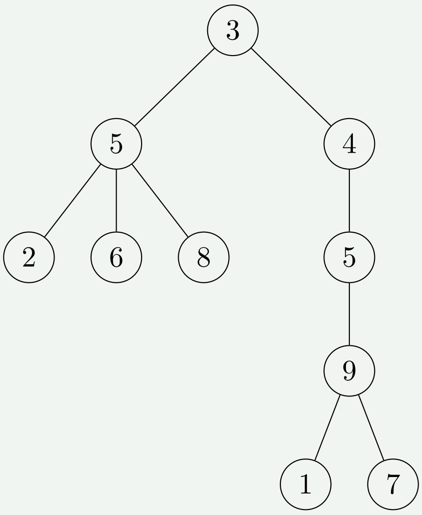
Every inversion of T is an inversion of either T_1 or T_2, so we have

$$\operatorname{inv}(T) = \widetilde{\operatorname{inv}}(T_1) + \operatorname{inv}(T_2),$$

where $\widetilde{\operatorname{inv}}(T_1)$ is again defined as the number of pairs (i, j) such that $i < j$ and j is on the shortest path from i to r – the difference is that r is not the minimal vertex of the tree, so it can now be involved in inversions.

Example 10.46

Find $\tilde{\text{inv}}(T_1)$ for the following tree T_1:



Solution. The inversions are $(1,9)$, $(1,4)$, $(1,3)$, $(2,5)$, $(2,3)$, and $(7,9)$, so there are 6 inversions. (Note that the labels inside T_1 are not necessarily consecutive, although it doesn't make a difference.) □

Now we have

$$I_n(x) = \sum_{k=1}^{n} \binom{n-1}{k-1} \tilde{I}_{k-1}(x) I_{n-k}(x).$$

The binomial coefficient counts the number of ways to distribute the vertices between T_1 and T_2 (we need to place 1 in T_1, as well as $k-1$ additional vertices which aren't 0 or 1). Here $I_{n-k}(x)$ is the usual inversion polynomial for T_2, and $\tilde{I}_{k-1}(x)$ is the modified inversion polynomial for T_1 (where we sum over all rooted trees on k vertices, instead of just trees rooted at the minimal vertex).

Now we want to relate $\tilde{I}_{k-1}(x)$ to the usual inversion polynomial $I_{k-1}(x)$.

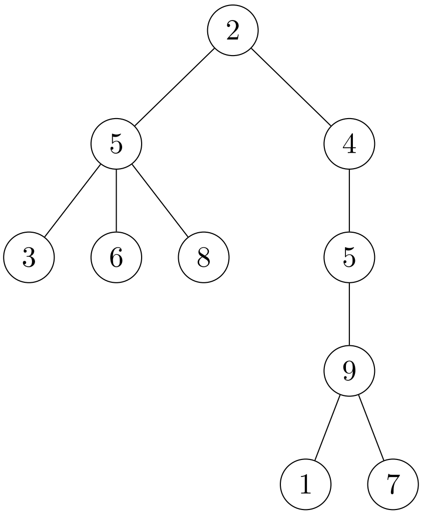
Lemma 10.47

If we sum over all trees T_1 on vertices $\{1,2,\ldots,k\}$ with root r, then we have

$$\sum_{T_1} x^{\tilde{\text{inv}}(T_1)} = x^{r-1} I_{k-1}(x).$$

Proof. Induct on r – when $r = 1$ is the minimal vertex, then we get the usual inversion polynomial.

For the inductive step, assume $r > 1$. Then let T_1' be the tree obtained from T_1 by switching the labels of r and $r-1$:



Then the inversions of T_1' are exactly the same as inversions of T_1 (with $r-1$ and r swapped), except that $(r-1, r)$ is no longer an inversion. This means

$$\sum_{T_1} x^{\tilde{\text{inv}}(T_1)} = x \sum_{T_1'} x^{\tilde{\text{inv}}(T_1')},$$

so by induction we're done. □

Now summing over all possible roots, we get

$$\tilde{I}_{k-1}(x) = (1 + x + x^2 + \cdots + x^{k-1}) I_{k-1}(x),$$

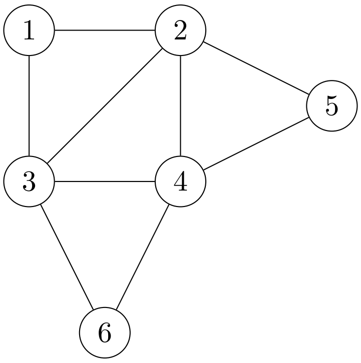
which gives the desired recurrence. □

This recurrence can be used to show the relation to parking functions given in Theorem 10.37 as well – we can show that the generating function for parking functions follows the same recurrence.

§11 Eulerian Walks

Definition 11.1. Given a graph G, an *Eulerian walk* is a path which uses every edge exactly once.

We'll focus on *closed* Eulerian walks, which begin and end at the same vertex.

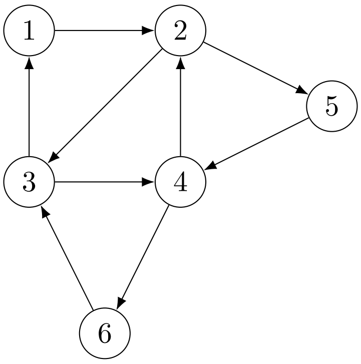


For example, $2 \to 4 \to 3 \to 1 \to 2 \to 5 \to 4 \to 6 \to 3 \to 2$ is a closed Eulerian walk on this graph.

Theorem 11.2 (Euler 1736)
A graph G has a closed Eulerian walk iff $\deg(v)$ is even for all vertices v, and G is connected.

(If we drop the requirement that the path must be closed, then we get the same condition, except that at most two degrees can be odd.)

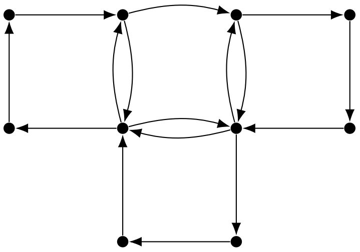
We can define Eulerian walks for directed graphs (called *digraphs*) as well, in the same way.



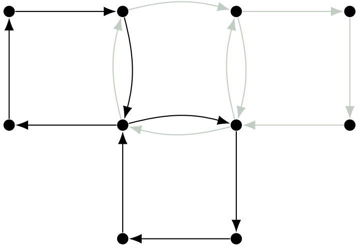
For example, $2 \to 3 \to 1 \to 2 \to 5 \to 4 \to 6 \to 3 \to 4 \to 2$ is a closed Eulerian walk on the above graph.

Theorem 11.3
A directed graph G has a closed Eulerian walk iff G is connected as an undirected graph, and $\operatorname{indeg}(v) = \operatorname{outdeg}(v)$ for all vertices v.

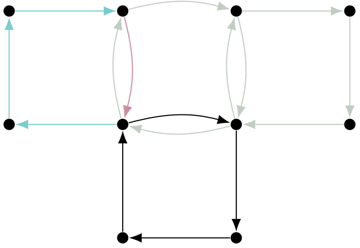
Proof. We won't formally prove this, but it's intuitively clear. For example, take the following graph:



Start at an arbitrary vertex, and keep walking until we get stuck. Since all in-degrees equal out-degrees, we can only get stuck at the starting vertex (for every other vertex, we've entered it more times than we've left, so there must be some unused outgoing edge). So this gives us some closed path:



Then if we haven't used every edge, there is some vertex in our path with an unused edge going out from it (because the graph is connected). Start at that vertex, and perform the same walking process (using the edges that aren't already in our path).



Then we can join the two paths at that vertex. Continue doing this until we've drawn every edge into our path; this gives a closed Eulerian walk. □

§11.1 The BEST Theorem

Question 11.4. If a digraph G satisfies these conditions, how many Eulerian walks does it have?

We need to be a bit careful of how we define the number of Eulerian walks – if we have a four-cycle with edges labelled a, b, c, and d, then (a, b, c, d) and (b, c, d, a) are both Eulerian walks. But they're essentially the *same* walk – they're obtained from each other just by cyclically shifting the edges. So we'll actually count closed Eulerian walks up to *cyclic shifts* – we can think of this as fixing the starting edge. (If we wanted to count these walks as distinct, we could just multiply by the number of edges.)

Theorem 11.5 (BEST Theorem)

If $G = (V, E)$ is a connected digraph where $\operatorname{indeg}(v) = \operatorname{outdeg}(v)$ for all $v \in V$, then the number of closed Eulerian walks (up to cyclic shifts) is

$$\operatorname{Arb}_r(G) \cdot \prod_{v \in V} (\operatorname{outdeg}(v) - 1)!\,,$$

for any $r \in V$.

Remark 11.6. BEST stands for the names of the authors – de Bruijn, von Aardenne-Ehrenfest, Smith, and Tutte. The theorem was proved in 1951.

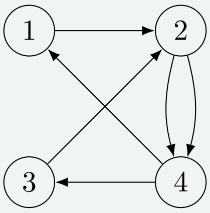
Recall that $\operatorname{Arb}_r(G)$ is the number of arborescences rooted at r, and is given by the cofactors of the Laplacian. Note that the above formula works for *any* root r, giving the following corollary:

Corollary 11.7

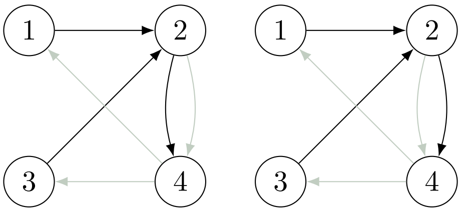
For a graph G satisfying the given conditions, $\operatorname{Arb}_r(G)$ does not depend on r.

Example 11.8

Consider the following graph:



Solution. There are two Eulerian walks ($1 \to 2 \to 4 \to 3 \to 2 \to 4 \to 1$, for either ordering of the edges $2 \to 4$). Meanwhile, all out-degrees are either 1 or 2, so all the factorials are 1. We can check that $\operatorname{Arb}_r(G) = 2$ for any root r. For example, for $r = 2$:



We can also count arborescences by using the Directed Matrix Tree Theorem: we have

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 2 & 0 & -2 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

(to count out-trees we place in-degrees on the diagonal, although in this case it doesn't matter). We can check that all cofactors are 2. □

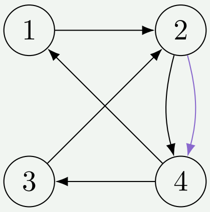
Remark 11.9. There is also a nice bijective proof that $\mathrm{Arb}_r(G)$ doesn't depend on r for such graphs.

Proof of Theorem 11.5. We'll construct a bijection between closed Eulerian walks, and objects consisting of an arborescence T rooted at r, and a permutation $\pi^{(v)}$ of size $\mathrm{outdeg}(v) - 1$ for each vertex $v \in V$.

First, fix a starting edge e which *ends* at r. Now for every vertex $v \neq r$, mark the *first* edge of the Eulerian walk to enter v.

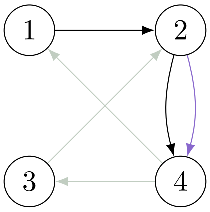
Example 11.10

Let G be the following graph, where $r = 4$ and e is the red edge.



Consider the Eulerian walk $2 \to 4 \to 3 \to 2 \to 4 \to 1 \to 2$ (starting with the red edge).

Solution. The first edge entering 1 in the walk is $4 \to 1$, the first edge entering 2 is $3 \to 2$, and the first edge entering 3 is $4 \to 3$. So we get the following marked edges:



□

Proposition 11.11

The marked edges form an arborescence rooted at r.

Proof. For every vertex except r, we mark exactly one edge entering it. So if there are no directed cycles, we can prove it's an arborescence.

Assume for contradiction that there is a cycle. This cycle can't contain r, since we didn't mark any edge entering r. Now let v be the first vertex in this cycle reached by the walk. Then we must have marked the edge into v by which we entered the cycle; so we couldn't have marked the edge entering v in the cycle, contradiction. □

So this shows how to obtain the arborescence; now we want to obtain the permutations. For every vertex v, let E_v be the set of edges entering v, except for the marked edges and the special edge e. Clearly $|E_v|= \mathrm{indeg}(v) - 1 = \mathrm{outdeg}(v) - 1$. So then take $\pi^{(v)}$ to be the permutation of E_v given by the order in which the edges appear in the Eulerian walk.

To see that this is a bijection, we want to show that given an arbitrary arborescence and collection of permutations, there exists a unique Eulerian walk corresponding to this data.

We can do this by working backwards: consider the last edge of the Eulerian walk (which occurs immediately before e). We know its endpoint v (which must be the starting point of e), and we know the order in which we take the edges entering v. So we choose the last edge in the permutation $\pi^{(v)}$.

We can continue doing this – when we visit a vertex for the second time (working backwards), we take its second-last edge, and so on (once we've used up all the edges in the permutation, we take the marked edge). This reconstructs the Eulerian walk. □

Remark 11.12. The arborescences come up from making sure that we really visit every edge. If we took any of the $\prod \text{outdeg}(v)!$ collections of permutations of *all* edges, then we could still work backwards, but it's possible that at some point, we'd get stuck (meaning that we close the walk without having reached every edge). The condition that the first edge of every permutation forms an arborescence is necessary and sufficient to prevent this from happening – in particular, this number is *at most* $\prod \text{outdeg}(v)!$.

§12 Graph Colorings

Let $G = (V, E)$ be an undirected graph.

Definition 12.1. For any nonnegative integer k, a k-coloring of G is a function $c : V \to [k]$ such that for any edge $(i, j) \in E$, we have $c(i) \neq c(j)$.

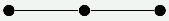
Example 12.2

Find the number of k-colorings of K_3.

Solution. We have k choices for the color of the first vertex, $k-1$ for the second, and $k-2$ for the third (since all colors must be distinct), giving $k(k-1)(k-2)$ colorings. □

Example 12.3

Find the number of k-colorings of the following graph:

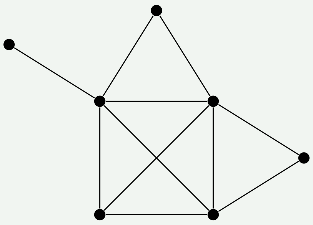


Solution. There are k choices for the color of the first vertex. Then there are $k-1$ choices for the second (which must be different from the first), and $k-1$ choices for the third (which must be different from the second). So the answer is $k(k-1)^2$. □

As we can see here, choosing a good order in which to color the vertices can make the problem much easier – if we colored the first vertex and then the third instead, then we'd have cases based on whether they were the same or different colors. As a more extreme example:

Example 12.4

Find the number of k-colorings of the following graph:



This graph looks much more complicated than the previous ones, but we'll see later that if we choose a good order to process its vertices in, then calculating its number of k-colorings is no harder than the previous examples.

§12.1 The Chromatic Polynomial

Lemma 12.5

For a fixed graph G, there exists a polynomial, called the *chromatic polynomial* $\chi_G(q)$, such that for any nonnegative integer k, the number of k-colorings of G is $\chi_G(k)$.

Proof. Induct on the number of edges. The idea we'll use here is *deletion-contraction*, which is a powerful tool in graph theory (as we'll see again later).

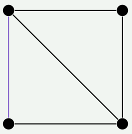
For the base case, if G has no edges, then the number of colorings is k^n (where n is the number of vertices). So $\chi_G(q) = q^n$.

For the inductive step, pick any edge $e \in E$ which is not a self-loop. (If the graph has a self-loop, then there are no k-colorings, so the chromatic polynomial is 0.)

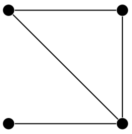
Then there are two things we can do with e: we can *delete* it (by erasing it from the graph) to get the graph $G \setminus e$, or *contract* it (by deleting it and gluing its endpoints together) to get the graph G/e.

Example 12.6

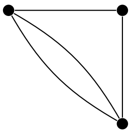
Consider the following graph, where e is the red edge:



Solution. If we delete e, we get the following graph:



Meanwhile, if we contract e, we glue together the left vertices:



Note that double edges don't affect the chromatic polynomial, so we could erase one without changing the chromatic polynomial, if we wanted to. □

Claim 12.7 — The number of k-colorings of G is the number of k-colorings of $G \setminus e$, minus the number of k-colorings of G/e.

Proof. First, any k-coloring of G is also a k-coloring of $G \setminus e$. Meanwhile, a k-coloring of $G \setminus e$ is valid iff the two endpoints of e are different colors, so we want to subtract the colorings where they're the same color. But these are exactly the k-colorings of G/e. □

But $G \setminus e$ and G/e both have fewer edges than G, so by induction, they both have chromatic polynomials. So we get the *deletion-contraction recurrence*

$$\chi_G(q) = \chi_{G \setminus e}(q) - \chi_{G/e}(q),$$

which must be a polynomial as well. □

§12.2 Acyclic Orientations

Now that we know $\chi_G(q)$ is a polynomial, we can plug in *any* input (not just nonnegative integers):

Question 12.8. How many ways are there to color a graph with -1 colors?

It turns out that $\chi_G(-1)$ does count something interesting.

Definition 12.9. An *acyclic orientation* of a graph G is a way to orient all edges of G, such that there are no directed cycles. Let $\mathrm{AO}(G)$ denote the number of acyclic orientations of G.

Example 12.10

Find $\mathrm{AO}(K_3)$.

Solution. There are 2^3 possible orientations, and exactly two are not acyclic (the cases where the edges form a clockwise or counterclockwise cycle). So $\mathrm{AO}(K_3) = 2^3 - 2 = 6$. □

Clearly $\mathrm{AO}(K_2) = 2$, so it appears that $\mathrm{AO}(K_n) = n!$ for all n. This is true – acyclic orientations of K_n correspond to orderings of the vertices (where we pick an ordering, and then direct all edges from left to right).

Theorem 12.11 (Stanley 1973)

If G has n vertices, then $\mathrm{AO}(G) = (-1)^n \chi_G(-1)$.

For example, we saw that $\chi_{K_3}(q) = q(q-1)(q-2)$, so the theorem gives $\mathrm{AO}(K_3) = (-1)^3 \cdot (-1)(-2)(-3) = 6$.

Proof. We know that the right-hand side satisfies a deletion-contraction recurrence, so it suffices to show that the left-hand side satisfies the same recurrence.

Lemma 12.12

We have

$$\mathrm{AO}(G) = \mathrm{AO}(G \setminus e) + \mathrm{AO}(G/e).$$

Meanwhile, we previously showed that

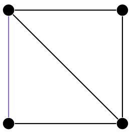
$$\chi_G(q) = \chi_{G\setminus e}(q) - \chi_{G/e}(q).$$

The discrepancy in sign comes from the $(-1)^n$, and the fact that contracting an edge deletes a vertex – if we let $\tilde{\chi}_G(q) = (-1)^n \chi_G(q)$, then we get the exact same recurrence

$$\tilde{\chi}_G(q) = \tilde{\chi}_{G\setminus e}(q) + \tilde{\chi}_{G/e}(q),$$

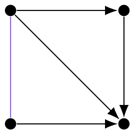
and it's easy to check that the theorem is true in the base case (where the graph has no edges).

Proof of Lemma 12.12. As an example, consider the following graph:

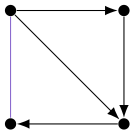


Take an acyclic orientation $\mathcal{O}$ of $G \setminus e$. Then to extend $\mathcal{O}$ to an orientation of G, we need to orient the edge e; there are two possible orientations. Let $e = (u, v)$.

Both orientations are valid if $\mathcal{O}$ does not contain a directed path between u and v. For example:

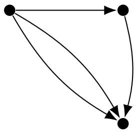


Meanwhile, if $\mathcal{O}$ does contain a directed path from u to v, then we must direct e as $u \to v$ as well:

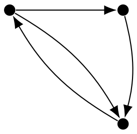


Note that at least one choice of orientation *must* be valid – otherwise $\mathcal{O}$ would contain a directed path from u to v, as well as a directed path from v to u; this would mean $\mathcal{O}$ was not acyclic to start with. So there's always one or two ways to orient e – there's two ways if $\mathcal{O}$ doesn't contain a directed path between u and v, and one way if it does.

But in the first case, when we contract e, we get an acyclic orientation of G/e, since a cycle would correspond to a directed path between u and v:



Meanwhile, in the second case, we end up with an orientation of G/e which *does* have a cycle (since the directed path between u and v is turned into a cycle):



So in the first case, an acyclic orientation of $G \setminus e$ corresponds to two acyclic orientations of G, and one of G/e. In the second, an acyclic orientation of $G \setminus e$ corresponds to one acyclic orientation of G, and zero of G/e. We have $2 = 1 + 1$ and $1 = 1 + 0$, so each acyclic orientation of $G \setminus e$ has the same contribution to both sides. □

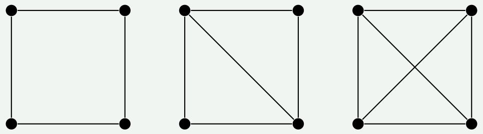
Then $\text{AO}(G)$ and $\tilde{\chi}_G(-1)$ satisfy the same recurrence, so we can show they are equal by induction. □

§12.3 Chordal Graphs

Definition 12.13. A graph G is *chordal* if every cycle in G of length at least 4 has a chord.

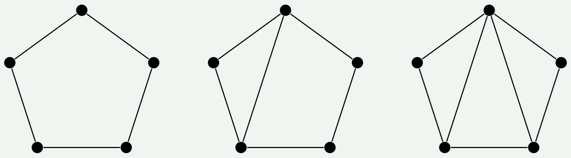
Example 12.14

A square is not chordal, but a square with one or both diagonals is chordal:

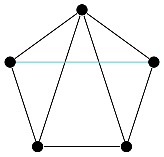


Example 12.15

A pentagon is not chordal, and if we add one edge, it's still not chordal. But if we add another edge inside the 4-cycle, then it becomes chordal:



Note that it's possible to start with a chordal graph and add an edge, and get a non-chordal graph. For example, we can add the blue edge to the chordal graph in the above example:



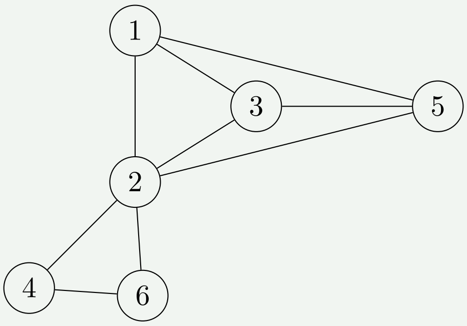
This graph is no longer chordal, since the 4-cycle formed by the bottom four vertices doesn't have a chord.

Definition 12.16. A *perfect elimination ordering* of a graph $G = (V, E)$ is a total ordering $v_1, \ldots, v_n$ of V, such that for all $i \geq 2$, the set $\{v_j \mid j < i \text{ and } (v_i, v_j) \in E\}$ is a clique.

A *clique* is a subgraph where any two vertices are adjacent.

Example 12.17

The following is a perfect elimination ordering:



Essentially, we start with the first vertex v_1, and keep adding vertices, such that the set we connect them to forms a clique. For example, here we start with 1, then add 2 and connect it to 1, then add 3 and connect it to 1 and 2 (which form a clique), then add 4 and connect it to 2, then add 5 and connect it to 1, 2, and 3 (which form a clique), then add 6 and connect it to 2 and 4 (which form a clique).

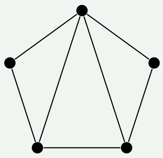
Theorem 12.18 (Fulkerson-Gross 1965)

A graph G is cordal iff G has a perfect elimination ordering.

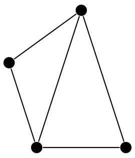
It's not hard to show that if G has a perfect elimination ordering, it must be chordal. The other direction is harder.

Example 12.19

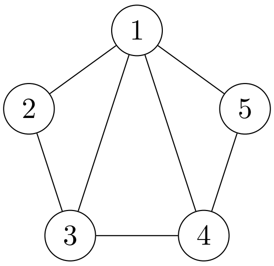
Show that our 5-vertex example graph has a perfect elimination ordering:



Solution. We can work backwards: first, we need v_5 to be connected to a clique, so we can let v_5 be the rightmost vertex. Then we can delete it:



Then we can pick v_4 to be another vertex which is now connected to a clique, and so on. For example:



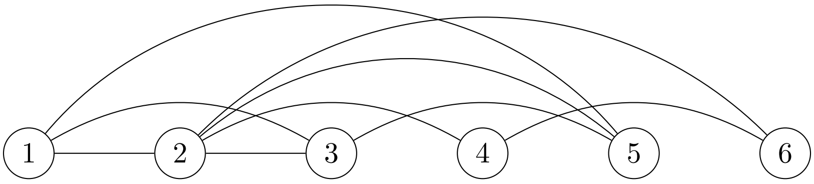
This gives one perfect elimination ordering of the graph. □

For a perfect elimination ordering $v_1, \ldots, v_n$, let $a_i = \#\{j < i \mid (v_i, v_j) \in E\}$, and consider the sequence of nonnegative integers $(a_1, a_2, \ldots, a_n)$ (where $a_1 = 0$).

Example 12.20

Find the sequence $(a_1, \ldots, a_n)$ for the graph in Example 12.17.

Solution. We can write out the graph horizontally:



Then a_i is the number of edges from v_i which go left, so we get the sequence $(0, 1, 2, 1, 3, 2)$. □

Theorem 12.21

For a chordal graph G corresponding to a sequence $(a_1, a_2, \ldots, a_n)$, we have

$$\chi_G(q) = \prod_{i=1}^{n}(q - a_i).$$

Corollary 12.22

For a chordal graph G, we have

$$\text{AO}(G) = \prod_{i=1}^{n}(a_i + 1).$$

So chordal graphs have simple formulas for the chromatic polynomial and number of acyclic orientations. For example, in our example graph we have

$$\chi_G(q) = q(q-1)(q-2)(q-1)(q-3)(q-2) = q(q-1)^2(q-2)^2(q-3).$$

Proof of Theorem 12.21. We can use essentially the same argument as we used for the complete graph – color vertices one by one, in the order given by the perfect elimination order.

For example, in the above example, there's k ways to color v_1. Then there's $k - 1$ ways to color v_2 (since it can be any color except the one used for 1), $k - 2$ ways to color v_3 (any color except the ones used for 1 and 2), $k - 1$ ways to color v_4 (any color except the one used for 2), and so on.

This works for any chordal graph. The reason it doesn't work for an arbitrary graph is because if v_3 is connected to both v_1 and v_2, then we need to choose a color for v_3 other than the ones used for v_1 and v_2, but it's possible that these colors are the *same* color (in which case only one color isn't allowed, instead of two). But the condition that the previous vertices form a clique guarantees that this doesn't happen – there's an edge between all vertices that v_i is connected to, so they must all be distinct colors. □

The large graph mentioned earlier, in Example 12.4, is actually chordal; so we can find a perfect elimination ordering, and easily calculate its number of k-colorings.

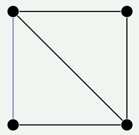
> **Remark 12.23.** It's possible for a graph G to have multiple perfect elimination orderings, which may give different sequences. But by this theorem we know that the multiset $\{a_1, \ldots, a_n\}$ is fixed – the chromatic polynomial doesn't depend on the ordering, so this multiset must be the same for any two perfect elimination orderings as well.

§12.4 More About Deletion-Contraction

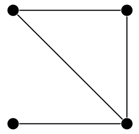
We've seen several graphical invariants which satisfy the deletion-contraction recurrence. There are others as well. For example, the number of spanning trees also satisfies deletion-contraction – deleting e corresponds to spanning trees which don't contain e, while contracting e corresponds to spanning trees which contain e. The number of subforests also satisfies deletion-contraction.

Example 12.24

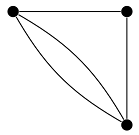
Consider deletion-contraction for the number of spanning trees of our usual graph:



Solution. We've seen before that this graph has 8 spanning trees. If we delete e, we get the following graph:



This graph has 4 spanning trees. Meanwhile, if we contract e, we get the following graph:



This graph has 4 spanning trees as well, and we have $8 = 4 + 4$. □

Question 12.25. Is there a universal deletion-contraction invariant of graphs?

Here *invariant* refers to a property which doesn't depend on the labelling of the vertices. By *universal*, we mean the most general – so any other deletion-contraction invariant should be a specialization of it. For example, we saw that $\mathrm{AO}(G)$ is a specialization of $\chi_G(q)$; we might ask whether there's a more general polynomial (in more variables) which encompasses all these invariants.

The answer is yes – there exists a polynomial in just two variables, called the *Tutte polynomial* $T_G(x, y)$. All other polynomials are obtained from the Tutte polynomial in some way – for example, by specializing the values, taking derivatives, and so on.

The Tutte polynomial should have some properties:

- It should satisfy deletion-contraction.
- It should be an invariant (meaning it doesn't depend on our way of labelling vertices or edges).
- It should have positive integer coefficients.

There are at least three different ways to define the Tutte polynomial; these ways each make some properties obvious, but not others. So in the proof, we'd define it in three different ways, and show that each definition gives the same polynomial.

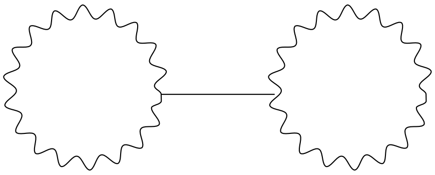
One definition is to define it by deletion-contraction:

Theorem 12.26 (Tutte)

For a graph $G = (V, E)$, there exists a unique polynomial $T_G(x, y)$ such that:

- For any edge $e \in E$ which is not a loop or a bridge, we have $T_G = T_{G\backslash e} + T_{G/e}$.
- If G has a bridges and b loops, and no other edges, then $T_G(x, y) = x^a y^b$.

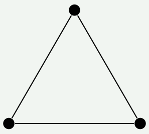
An edge is a *bridge* if deleting it would increase the number of connected components.



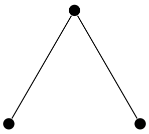
So deletion-contraction holds by definition. By induction, the polynomial must have positive integer coefficients. It's also clear that the polynomial is an invariant (since we didn't reference any labelling of vertices or edges). So this definition makes it clear that the polynomial satisfies all three properties. But what *isn't* obvious is that the polynomial actually exists – it's *a priori* not obvious that any order of deletion and contraction should give us the same result. So using this definition, the claim of existence is the only thing that needs to be shown. (We won't have time to prove it here.)

Example 12.27

Find the Tutte polynomial of K_3.

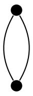


Solution. Take any edge. If we delete it, we get a carrot:



This has two bridges and no loops, so its Tutte polynomial is x^2.

Meanwhile, if we contract the edge, then we get a 2-cycle:

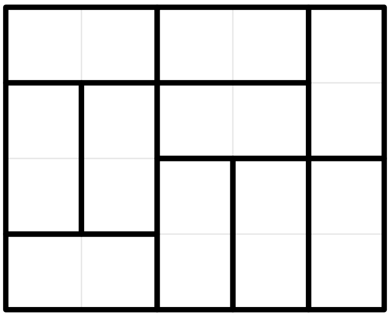


Pick either edge. If we delete it, then we get a single edge (which contributes x), while if we contract it, then we get a loop (which contributes y). So the Tutte polynomial of this graph is $x + y$. Combining these, we get $T_{K_3}(x, y) = x^2 + x + y$. □

Remark 12.28. In this example, it's clear that the order in which we choose edges doesn't matter. But this is more mysterious for larger graphs – that no matter what order we delete and contract the edges in, we end up with the same result.

§13 Domino Tilings

Take a $n \times m$ rectangle. We want to subdivide it into dominoes (which are 2×1 rectangles), and we are interested in the number of ways to do so.

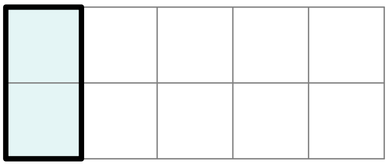


Example 13.1

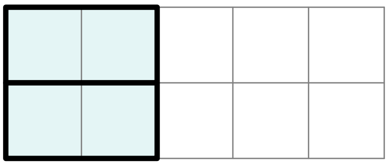
The number of domino tilings of a $2 \times n$ rectangle is F_{n+1} (the $(n+1)$th Fibonacci number).

Proof. Let a_n be the number of tilings. We want to show that a_n satisfies the same recurrence as the Fibonacci numbers.

Consider the square in the upper-left corner. If it is part of a vertical domino, then we're left with a $2 \times (n-1)$ rectangle, giving a_{n-1} ways.



Meanwhile, if it's part of a horizontal domino, then the lower-left corner must also be part of a horizontal domino. So we're left with a $2 \times (n-2)$ rectangle, giving a_{n-2} ways.

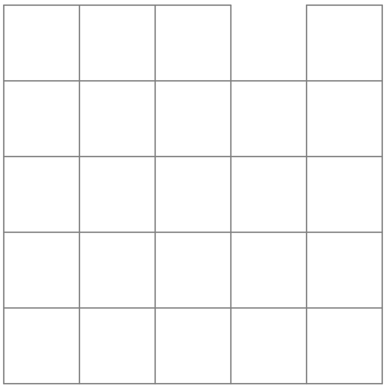


So we get $a_n = a_{n-1} + a_{n-2}$, which is the Fibonacci recurrence. It's easy to check that the initial conditions hold, so $a_n = F_{n+1}$ by induction. □

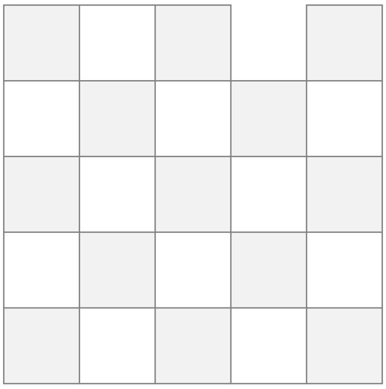
So in some sense, we can think of the number of domino tilings of a $m \times n$ rectangle as a generalization of Fibonacci numbers.

There are some obvious constraints: the number of tilings of a 5×5 square is 0, since there's an odd number of squares. So in order for a $m \times n$ rectangle to have tilings, either m or n must be even.

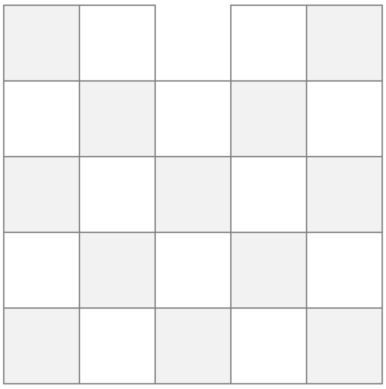
Given a 5×5 square, in order to make the number of squares even, we can erase one box:



But if we erase a box in the above way, there's still 0 ways: color the grid in a checkerboard pattern. Then every domino has one black and one white square, but this grid doesn't have the same number of black and white squares.



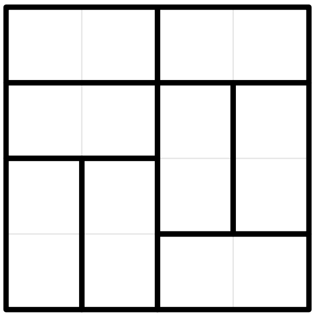
So in order to have a nonzero number of tilings, we need to remove a square of the same color as the corners:



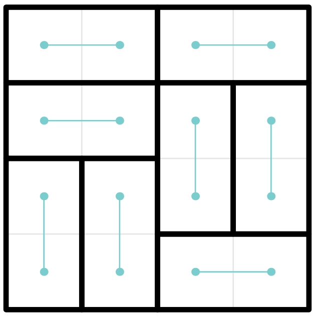
Now there is a nonzero number of tilings, and we'll see later how to count them.

§13.1 Perfect Matchings

Domino tilings are related to perfect matchings. For example, consider the following tiling:

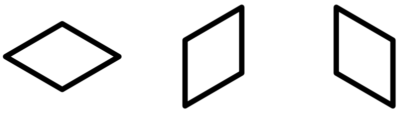


We can draw a dot at the center of each square, and connect two dots iff they're in the same domino:

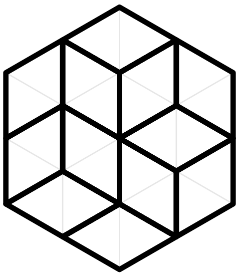


So counting the number of domino tilings is the same as counting the number of perfect matchings of a grid graph.

Another interesting class of tilings is *rhombus* tilings – suppose we have a regular hexagon of side length n, which we want to subdivide into rhombi. There are three possible orientations of each rhombus:

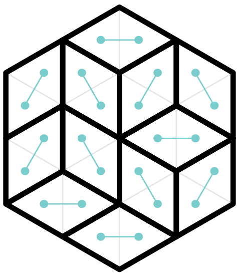


For example, if $n = 2$:

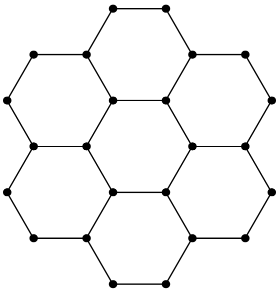


This picture looks like a pile of boxes – so rhombus tilings are related to a three-dimensional generalization of Young diagrams, called *plane partitions*.

We can think of rhombus tilings as perfect matchings as well: again, draw a dot at the center of each triangle, and connect dots in the same rhombus.



This now gives a perfect matching of the *honeycomb* graph (a hexagonal lattice):



So domino tilings correspond to perfect matchings of the grid graph, and rhombus tilings to perfect matchings of the honeycomb graph.

§13.2 Counting Tilings

People have developed many different techniques for calculating the number of such tilings. There is an explicit formula for the number of rhombus tilings:

Theorem 13.2 (MacMahon 1898)

If we have a hexagon of side lengths m, n, and k, then the number of rhombus tilings of the hexagon is

$$\prod_{a=1}^{m}\prod_{b=1}^{n}\prod_{c=1}^{k}\frac{a+b+c-1}{a+b+c-2}.$$

Note that it's not *a priori* obvious that this expression is an integer. We don't have time to prove the theorem, but it has several nice proofs.

There is also an explicit formula for the number of domino tilings. This was proved much later, and by very different methods:

Theorem 13.3 (Kastelyn 1961)

Let m and n be integers, with n even. Then the number of domino tilings of a $n \times m$ rectangle is

$$\prod_{k=1}^{n/2}\prod_{\ell=1}^{\lfloor m/2\rfloor}\left(4\cos^2\frac{\pi k}{n+1}+4\cos^2\frac{\pi \ell}{m+1}\right).$$

It's even more miraculous that this expression is an integer – the individual terms are irrational numbers, but when we multiply all of them, we get the number of domino tilings.

Remark 13.4. This formula can be proved by doing calculations with matrices. The *permanent* of a matrix is a function defined similarly to the determinant, but instead of adding $(-1)^{\text{sgn}(\sigma)}$ times the product of the entries $a_{i\sigma(i)}$, we just add the product directly. This can be used to count the number of perfect matchings of a bipartite graph. There's another function on matrices related to the determinant, called the *Pfaffian*. In planar graphs, it's possible to relate the permanent and Pfaffian, and perform calculations with the eigenvalues to get this formula – these strange expressions come from the eigenvalues of a matrix.

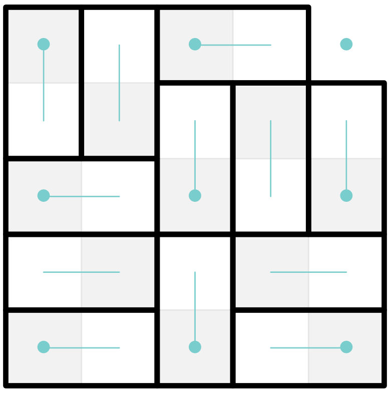
Now we'll consider the case mentioned earlier, where we have a rectangle with odd dimensions, and one box removed from the border. It turns out that there is a formula for this configuration as well:

Theorem 13.5

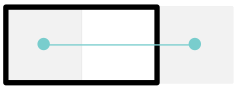
Let $m = 2k - 1$ and $n = 2\ell - 1$, and let R be the $m \times n$ rectangle with one box a deleted, such that a is on the boundary of the rectangle and has the same color as the corners of the rectangle.

Then the number of domino tilings of R is the number of spanning trees of the $\ell \times k$ grid graph.

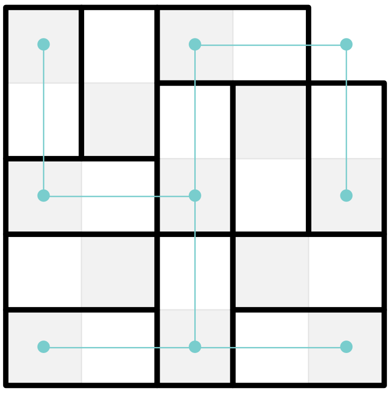
Proof. We'll find a bijection between domino tilings of R and spanning trees of the grid graph.



Draw a large dot in each shaded square in an odd row and column. Then we'll connect some of these large dots: for each domino with a large dot, connect this dot to the large dot on the other side of the domino:



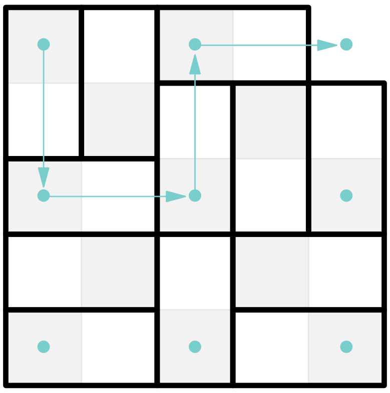
For example, in the above domino tiling, we get:



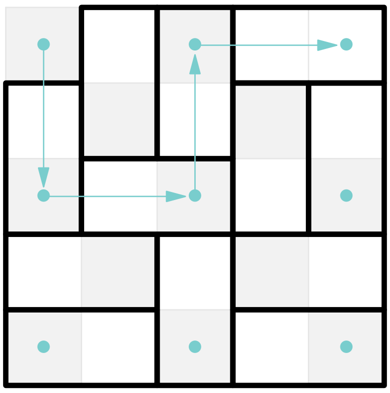
We can check that this always produces a spanning tree on the large dots, and that every spanning tree produces one tiling. □

Note that this doesn't depend on which box was deleted – if we delete *any* box of the correct color from the border, we end up with the same number of domino tilings.

In fact, we can find a direct bijection between them: if the original deleted box was a and the new deleted box is b, then draw the path from b to a, and shift all dominoes one step along that path. For example, if we want to delete the top-left corner instead:



We get this path by starting at b, and then following the same rule (if we're at a large dot, then go to the large dot on the other side of its domino) until we reach a. Then we shift all dominoes one step forwards on that path:



This gives a domino tiling covering every square except b.

> **Remark 13.6.** This topic has many deep results; the ones we saw here are just the tip of the iceberg. But we don't have enough time to cover every topic in combinatorics.