18.212: Algebraic Combinatorics

Andrew Lin

Spring 2019

This class is being taught by **Professor Postnikov**.

April 10, 2019

We're going to continue with problem presentations today.

Problem 1

Show that the Bell numbers can be calculated using the Bell triangle:

$$\begin{array}{ccccc} & & & & 1 \\ & & & 1 & 2 \\ & & 2 & 3 & 5 \\ & 5 & 7 & 10 & 15 \\ 15 & 20 & 27 & 37 & 52 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

In this triangle, the first number in each row (except the first row) equals the last number in the previous row; and any other number equals the sum of the two numbers to the left and above it. The Bell numbers $B(0) = 1, B(1) = 1, B(2) = 2, B(3) = 5, B(4) = 15, B(5) = 52, \cdots$ appear as the first entries (and also the last entries) in rows of this triangle.

Solution by Fadi Atieh. We'll prove this by induction: the induction case is easy, because $B_0 = B_1 = 1$. Now looking at the induction step, assume that we have already built k rows of this triangle. We know that the leftmost entry is B_{k-1}, and the right-most entry is B_k, so the $k+1$th row begins with B_k. The next entry is $B_k + B_{k-1}$, and notice that we can recursively push our numbers up and to the left until we reach the left diagonal. We just need to figure out the actual recurrence relation!

Well, if we try to construct B_{k+1}, the rightmost entry in the $k+1$th row, we can sum it by taking some left and up steps to the diagonal. To get to B_i, we need i left steps and $k - i$ up steps, which means

$$B_{k+1} = \sum_{i=0}^{k} \binom{k}{i} B_i.$$

This is the recursive formula for bell numbers, because we pick some i numbers in our partition to be in the same partition as $k+1$. $\square$

Problem 2

Show that the Stirling numbers have the following recurrence relations:

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

and

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

Solution by Ramya Durvasula. $S(n,k)$ tells us how to place n numbers into a set-partition with k groups. If we partition $n+1$ into k groups, one way is to have $n+1$ on its own, in which case we partition 1 through n into $k-1$ groups: thus there are $S(n,k-1)$ ways to do this. On the other hand, $n+1$ may not be on its own, in which case we can partition the numbers 1 through n into k sets. Then we put $n+1$ into one of the k sets in one of k ways, so this process has $kS(n,k)$ possible choices.

Putting these together, $S(n+1,k) = kS(n,k) + S(n,k-1)$, as desired.

Similarly, $c(n,k)$ tells us how to make a permutation of n numbers with k cycles. If we are trying to compute $c(n+1,k)$, again, we think about where $n+1$ goes: if it goes in its own cycle, we have $c(n,k-1)$ ways to do this, since $n+1$ gets placed in its own cycle. On the other hand, if $n+1$ is in a cycle, first create k cycles with the numbers 1 through n: there are $c(n,k)$ ways to do this, and then we can add $n+1$ to one of the n spots after any element, so this process has $nc(n,k)$ ways.

Putting these together, $c(n+1,k) = nc(n,k) + c(n,k-1)$, as desired. ☐

Problem 3

We have operators $X : f(x) \to xf(x)$ and $D : f(x) \to f'(x)$. Defining $f_n(x) = (x+D)^n(1)$; for example, $f_0(x) = 1$, $f_1(x) = x$, $f_2(x) = x^2+1$, $f_3(x) = x^3+3x$. Find the constant term $f_n(0)$ in terms of n.

Solution by Ganatra. We know that recursively,

$$f_n(x) = xf_{n-1}(x) + f'_{n-1}(x)$$

so plugging in $x = 0$,

$$f_n(x) = f'_{n-1}(x).$$

The idea is that f_n is entirely dependent on the value of f_{n-1}. Writing out a few more values,

$$f_4(x) = x^4 + 6x^2 + 3, f_5(x) = x^5 + 10x^3 + 15x, f_6(x) = x^5 + 15x^4 + 45x^2 + 15,$$

$$f_7(x) = x^7 + 21x^5 + 105x^3 + 105x, f_8(x) = x^8 + 28x^6 + 210x^4 + 420x^2 + 105.$$

We start to notice some patterns: x^n is always the leading coefficient of $f_n(x)$, and the last coefficients are 1, 1, 1, 3, 3, 15, 15, 105, 105. In fact, we can find a general formula

$$f_n(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} x^{n-2k} \frac{(n)_{2k}}{k!2^k}.$$

This can be proved with induction. So now

$$\binom{n}{k} = \frac{(n)_k}{k!} \implies (n)_k = \frac{n!}{(n-k)!}, (2k)!! = 2^k k!$$

Plugging these in, we can say that $(n)_{2k} = \frac{n!}{(n-2k)!}$. We're only concerned with the last term of our formula for f_n: if n is odd, our constant term is 0, and otherwise, our expression simplifies to (for n even) is

$$f_n(0) = \frac{n!}{n!!} = (n-1)!!.$$

☐

Professor Postnikov says that it turns out there is a relation between this problem and the previous problem about up and down operators on Young's lattice: the key relation is that $DU - UD = I$, $D(\varnothing) = 0$. These alone imply that the number of paths of length $2n$ is $(2n-1)!!$.

The coefficients $\frac{(n)_{2k}}{k!2^k}$ in $f_n(x)$ also tell us something about Young's lattice: it's the number of ways to take $2n$ steps to get to the kth level.

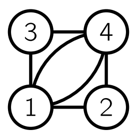
Time to move on to a new topic! We're going to discuss **spanning trees**.

Definition 4

Let $G = (V, E)$ be a graph with a set of vertices V and edges E with no loops (edges with both endpoints being the same vertex), but where multiple edges are allowed. Then a **spanning tree** of G is a **subgraph** T with the same set of vertices $T = (V, E')$, where $E' \subset E$ which is a tree that connects all vertices.

Notably, it has no cycles, and we can't have repeated edges either (since they form a cycle). So we're picking some $|V| - 1$ edges that "span" or connect all vertices of the original graph. Define $T(G)$ to be the number of spanning trees of a graph G.

For example, consider the graph



For this graph, $T(G) = 12$, because there are 4 ways to not use the double edge and $2 \cdot 2 \cdot 2$ ways to pick the double edge.

Fact 5

The number of spanning trees on K_n,

$$T(K_n) = n^{n-2}.$$

This is because we can pick any labeled tree!

It turns out there is a formula in general called the matrix-tree theorem!