LECTURE 16 – BIVARIATE ENUMERATION AND CATALYTIC VARIABLES

JAY PANTONE

CATALYTIC VARIABLES

We've previously seen the utility of bivariate generating functions in studying the structure of combinatorial objects. By adding in a new variable to track a particular feature of an object, one can obtain a wealth of information including moments of the distribution (expected value, variance, etc.) and whether that distribution is concentrated about the mean.

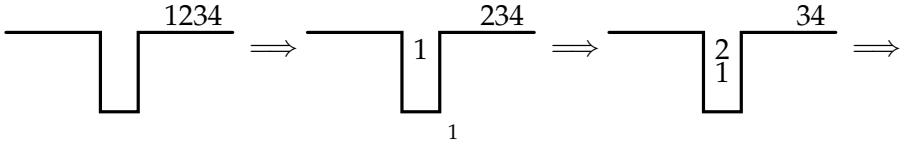
We study here a new application of bivariate generating functions: enumeration of counting sequences. Previously, we have found bivariate generating functions by taking a symbolic construction for the univariate case and adding markers. In these instances, we could find the counting sequence for the underlying objects with no need for the bivariate form. However, it is often the case that in order to find the (univariate) counting sequence for the class one must start with a more refined bivariate (or multivariate) functional equation which can them (sometimes) be solved to find the univariate generating function.

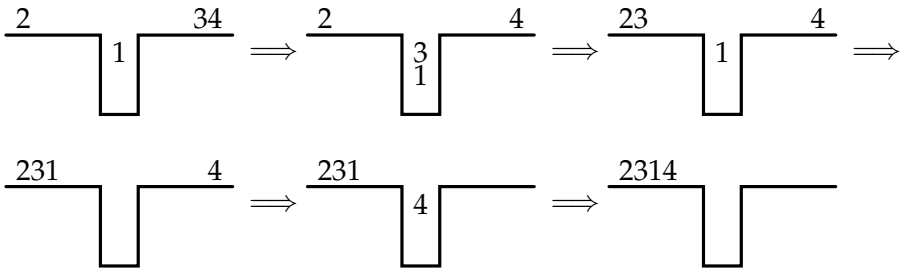
To illustrate this phenomenon, we will consider a class question from the field of pattern-avoiding permutations. This result appears in Knuth's *The Art of Computer Science* in Section 2.2.1, and is one of the earliest in the area.

Let π of length n be a permutation, thought of in one-line notation. Let π_i be the ith entry of π. For example, $\pi = 74356128$ is a permutation of length 8, and $\pi_4 = 5$. We say that π *contains* a smaller permutation σ of length $k \leq n$ if there exist indices $i_1 < i_2 < \ldots < i_k$ such that the entries $\{\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_k}\}$ are in the same relative order as σ. For instance, the permutation π above contains 321 in a number of places (e.g., 743, 762, etc.) but *avoids* 132 (i.e., there are no three entries of π that when left-to-right go "smallest, biggest, middle").

This pattern-containment relation induces a poset on the set of all permutations. A *permutation class* $\mathcal{C}$ is a downset in this poset. Any permutation class can be defined uniquely by the set of minimal permutations that it avoids, called its *basis*. The class with basis B is denoted $\mathrm{Av}(B)$.

A stack is an object from computer science that consists of a single column of storage into which one may push entries from an input list and from which one may pop entries to an output list. Knuth asked which permutations can be generating by feeding the identity permutation $12\cdots n$ into the stack. For example, the permutation 2314 can be formed by the following operation sequence.

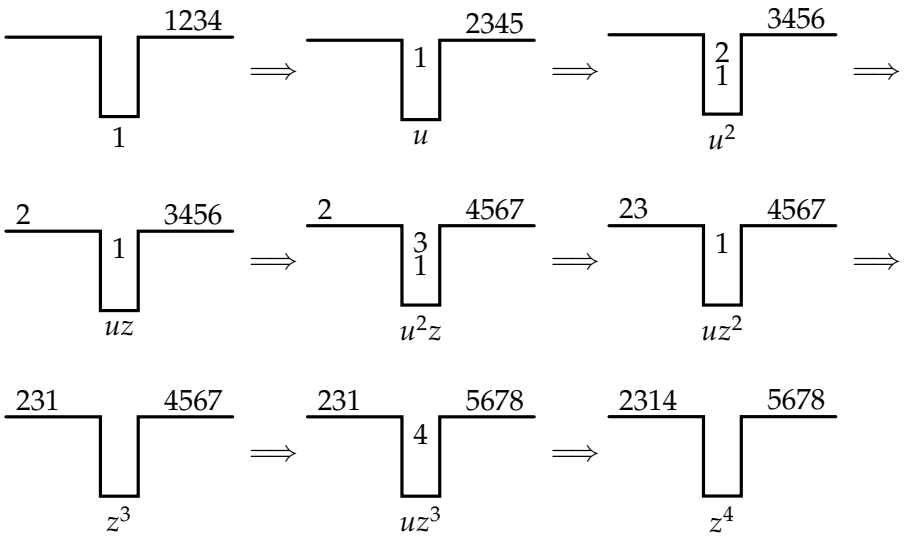




It is not hard to see that a permutation can be generated by a stack if and only if it avoids 312. To find the enumeration of this class $\mathrm{Av}(231)$, we will use the stack operations to set up a bivariate generating function. This is a great example of an instance where we cannot immediately find a univariate generating function, but by creating a more refined structural decomposition involving a second property tracked by u we can accomplish the same goal. In these situations, u is called a *catalytic variable*: we don't really care about what it tracks except that we need it in the first stage to get function equation, and then we eliminate it later to recover the univariate generating function.

Consider a stack with an infinite increasing permutation $123\cdots$ ready to be fed into the input. A permutation can be generated by the stack if there is some sequence of pushes and pops that leads to a complete permutation of size n in the output and nothing inside the machine. Critically, with such a simple machine every different push/pop sequence leads to a different permutation in the output.

Let $f(z,u)$ be the bivariate generating functions for states of the stack while using this infinite input sequence to generate permutations, where z tracks the number of pops that have been done (i.e., the length of the output queue), and u tracks the number of entries currently in the stack. The generating function $f(z,u)$ is simultaneously tracking all possible evolutions of the machine and all permutations that these lead to. To illustrate, let us repeat the steps in generating the permutation 2314 now stating the term in $f(z,u)$ that each state corresponds to.



The states of the machine that have a complete permutation in the output are exactly those whose u term has a power of zero. Thus if we can find the generating function $f(z,u)$ described above, then the univariate generating function for $\mathrm{Av}(312)$ is $f(z,0)$.

The construction of $f(z,u)$ relies on the following structural decomposition: every state of the machine is either

(1) the initial empty state,

(2) arises from pushing something into the stack, or

(3) arises from popping something from a *non-empty* state.

This immediately gives the functional equation

$$f(z,u) = \underbrace{1}_{\substack{\text{initial}\\ \text{state}}} + \underbrace{uf(z,u)}_{\substack{\text{push to}\\ \text{stack}}} + \underbrace{\frac{z}{u}(f(z,u) - f(z,0))}_{\substack{\text{pop from}\\ \text{nonempty state}}}.$$

On the face of it, this is a single equation with two unknowns ($f(z,u)$ and $f(z,0)$), but of course these two unknowns are inextricably linked. We will see shortly how this relationship can be exploited to solve for $f(z,0)$ using a technique known as the *kernel method*.

HOPS, STEPS, AND JUMPS

Acknowledgement

Much of the material in this section originates from discussions with David Bevan.

We now examine another class of combinatorial problems that admits a nice translation to bivariate generating functions (with a single catalytic variable). A *walk* on the half-line $[0,\infty)$ is a sequence of steps that starts at 0 and never becomes negative. A Dyck path, though often drawn as a two-dimensional object, is really just a walk on the half-line with legal step set $\{-1,1\}$.

Given any class of walks with a specified set of restrictions, we will consider $f(z,u)$ to be the bivariate generating function for the walks where z tracks the length of the walk and u tracks the final end-point. In keeping with the literature, we may refer to this final end-point as *height* or *altitude*. Every such $f(z,u)$ has the form

$$f(z,u) = f_0(z) + f_1(z)u + f_2(z)u^2 + \cdots,$$

where $f_i(z)$ is the generating function for walks in the class that end at i. In particular, $f_0(z)$ is the generating function for walks that end at the origin. Each $f_i(z)$ can be found from $f(z,u)$ by differentiating i times with respect to u, dividing by $i!$, and substituting $u = 0$.

We consider walks that are built from combinations of three types of moves: hops, steps, and jumps.

- A *hop* is a step from height k to any height in $\{0,1,\ldots,k-1\}$.

- A *step* of size s, for $s \geq 0$ is either:
  - an up-step from k to $k+s$, or
  - a down-step from k to $k-s$ (only allowed if $k \geq s$).
- A *jump to j* is a step from any height to j.

Many types of walks can be formed by taking a union of these allowed moves at each step, and concatenating these moves together. For example, Dyck paths are a special case of walks where the moves allowed at any point are a step of size $+1$ and a step of size -1 (as long as the current position is at least 1).[1]

In order to form the associated functional equations, we must determine the effect of each type of move on terms $a_{n,k}z^n u^k$. We shall start with the easiest such move: the up-step of size s.

Steps. For any current walk of length n at position k, taking an up-step of size s transforms $a_{n,k}z^n u^k$ into $a_{n,k}z^{n+1}u^{k+s}$. Of course, we want to operate on the level of generating functions, not term-by-term. To do this, we realize that we can enact this transformation on every terms simultaneously with the transformation

$$\mathsf{S}_{+s}[f(u,z)] = zu^s f(u,z).$$

For a down-step of size s, we must be careful not to let the walk pass below the origin. To ensure this, we only act on those terms of $f(z,u)$ for which the exponent of u is at least s. On the term level, this amounts to

$$a_{n,k}z^n u^k \longmapsto \begin{cases} 0, & k < s \\ a_{n,k}z^{n+1}u^{k-s}, & k \geq s \end{cases}$$

On the global generating function level, we can accomplish this with the transformation

$$\mathsf{S}_{-s}[f(u,z)] = zu^{-s}(f(z,u) - f_0(z) - f_1(z)u - \cdots - f_{s-1}(z)u^{s-1}).$$

Jumps. A jump to j is a move in which a walk travels from any height to height j. For each term of the generating function $f(z,u)$ this is equivalent to $a_{n,k}z^n u^k \mapsto a_{n,k}z^{n+1}u^j$. On the global level, this is

$$\mathsf{J}_j[f(u,z)] = zu^j f(z,1).$$

Substituting $u = 1$ into $f(z,u)$ has the effect of ignoring the position of the walks (or more precisely, of deleting the record of position). Then, the multiplication by u^j repositions every walk to height j.

One can also define a more refined version of jumps: A jump from i to j moves those walks at height i directly to height j. All other walks cease. Of course, in practice this would be combined with other allowed moves (e.g., perhaps every walk at height k can jump up to height $2k$; this is an infinite sum of refined jumps).

[1] These more general walks are the meanders that appeared on a recent homework assignment.

Hops. A hop (or, a *hop down*) is a move in which any walk at height k can fall to any altitude in $\{0, 1, \ldots, k-1\}$. On the term level, this equates to

$$a_{n,k} z^n u^k = a_{n,k} z^{n+1} \left(u^0 + u^1 + u^2 + \cdots + u^{k-1} \right) = a_{n,k} z^{n+1} \frac{1-u^k}{1-u}.$$

This is a bit trickier to enact on the level of generating functions. It helps to think of the right-hand side of the above equation as

$$\frac{z}{1-u} \left(a_{n,k} z^n - a_{n,k} z^n u^k \right).$$

We can now see that a hop equations to the generating function transformation

$$\mathsf{H}[f(z,u)] = \frac{z}{1-u} \left(f(z,1) - f(z,u) \right).$$

Before we dive into how we can build classes of walks with these operators, we should point out something interesting about the transformation for hops.

Consider a bivariate generating function $g(z,u) = \sum_{n,k \geq 0} g_{n,k} z^n u^k$. Assume that $g(z,1)$ is a valid formal power series in z and let g_n be the coefficient of z^n in $g(z,1)$. We see by algebraic manipulation that

$$\begin{aligned}
\left[\frac{g(z,1) - g(z,u)}{1-u} \right]_{u=1} &= \left[\frac{\sum_{n \geq 0} g_n z^n - \sum_{n,k \geq 0} g_{n,k} z^n u^k}{1-u} \right]_{u=1} \\
&= \left[\sum_{n \geq 0} \left(\frac{g_n - \sum_{k \geq 0} g_{n,k} u^k}{1-u} \right) z^n \right]_{u=1} \\
&= \left[\sum_{n \geq 0} \left(\frac{\sum_{k \geq 0} g_{n,k} - \sum_{k \geq 0} g_{n,k} u^k}{1-u} \right) z^n \right]_{u=1} \\
&= \left[\sum_{n \geq 0} \left(\frac{\sum_{k \geq 0} g_{n,k} (1-u^k)}{1-u} \right) z^n \right]_{u=1} \\
&= \left[\sum_{n \geq 0} \left(\sum_{k \geq 0} \frac{g_{n,k} (1-u^k)}{1-u} \right) z^n \right]_{u=1} \\
&= \left[\sum_{n,k \geq 0} g_{n,k} (1 + u + \cdots + u^{k-1}) z^n \right]_{u=1} \\
&= \sum_{n,k \geq 0} k g_{n,k} z^n \\
&= g_u(z,1).
\end{aligned}$$

As this computation results in the derivative of g with respect to u, it is common to call the expression

$$\frac{g(z,1) - g(z,u)}{1-u}$$

the *discrete derivative with respect to* u. We will soon see the composition of hops results in these derivatives appearing in our functional equations.

Crafting Walks. We will consider a few examples of walks whose bivariate generating functions can be easily determined by the operations above. Although these are all described as walks, they have combinatorial applications to many different types of objects and so are very broadly applicable.

Example: Consider the class of walks in which every point the height of the walk can stay the same, increase by one, or decrease by one. This corresponds to a step set of $\{-1, 0, 1\}$, though we should note that our operators work in more general cases where the step set is not fixed. As such we recover the form

$$f(z,u) = 1 + \mathsf{S}_{-1}[f(z,u)] + \mathsf{S}_{+0}[f(z,u)] + \mathsf{S}_{+1}[f(z,u)],$$

where the term 1 accounts for the initial walk of length and height 0. Applying the transformations for each operator, we get

$$f(z,u) = 1 + \frac{z}{u}(f(z,u) - f(z,0)) + zf(z,u) + zuf(z,u).$$

Although we have not yet seen how to solve equations of this form, the univariate generating functions for these walks counted by length, $f(z,u)$ is easily recovered by using the kernel method.

Example: For the next example, consider the class of walks that may increase their altitude by at most 1 per step, may stay at the same height, or may hop down to any height between 0 and the current height. Then,

$$f(z,u) = 1 + \mathsf{S}_{+1}[f(z,u)] + \mathsf{S}_{+0}[f(z,u)] + \mathsf{H}[f(z,u)],$$

which gives the functional equation

$$f(z,u) = 1 + zuf(z,u) + zf(z,u) + \frac{f(z,1) - f(z,u)}{1-u}.$$

The kernel method can be used to solve for $f(z,1)$, and then since the equation for $f(z,u)$ is linear in $f(z,1)$ we can solve for $f(z,u)$.

Example: In this example we obtain an equation that cannot be solved by the kernel method (at least not in an elementary way). We study walks with the step set $\{-2,-1,0,1,2\}$. These satisfy

$$f(z,u) = 1 + \mathsf{S}_{-2}[f(z,u)] + \mathsf{S}_{-1}[f(z,u)] + \mathsf{S}_{+0}[f(z,u)] + \mathsf{S}_{+1}[f(z,u)] + \mathsf{S}_{+2}[f(z,u)],$$

and so

$$\begin{aligned} f(z,u) &= 1 + \frac{z}{u^2}(f(z,u) - f_0 - uf_1) + \frac{z}{u}(f(z,u) - f_0) + zf(z,u) + zuf(z,u) + zu^2 f(z,u) \\ &= 1 + \frac{z}{u^2}((1+u+u^2+u^3+u^4)f(z,u) - (1+u)f_0(z) - uf_1(z)). \end{aligned}$$

Recalling that $f_0(z) = f(z,0)$ and $f_1(z) = uf_u(z,0)$, we have

$$f(z,u) = 1 + \frac{z}{u^2}((1+u+u^2+u^3+u^4)f(z,u) - (1+u)f(z,0) - uf_u(z,0)).$$

This is a single equation with three (linked) unknowns. Instead of the kernel method, we will use a (rigorous!) method that we call *guess-and-check*.

Example: Our last example shows how walks can be used to count other combinatorial objects. Consider the set of permutations of length $2n$ that avoid a 321 pattern, such that each of the n adjacent pairs contain an increasing pair of entries.[2]

The translation to walks is as follows. An inversion in a permutation is a pair of indices $i < j$ such that $\pi_i > \pi_j$. Given a permutation π, let $k(\pi)$ be the number of entries of π whose value is greater than that of the rightmost point of π that is part of an inversion. To construct a permutation of length $2n+2$ from a permutation of length $2n$, we insert two entries to right-hand end of π. They must be in increasing order to satisfy the definition of this class. We must determine how k is affected by these insertions. For brevity we leave the reader to verify that the allowable transitions of k while inserting these two entries are satisfied by

$$f(z,u) = 1 + \mathsf{H}_1[\mathsf{H}_1[f(z,u)] + \mathsf{H}_2[f(z,u)] + \mathsf{S}_{+2}[f(z,u)],$$

where $\mathsf{H}_i = S_{+i} \circ \mathsf{H}$, we can be imagined as a hop from k to $\{i, i+1, \ldots, k+i-1\}$. This results in a functional equation of the form

$$f(z,u) = 1 + \frac{z}{(1-u)^2}(u^2(1-u+u^2)f(z,u) - u^3 f(z,1) + u(1-u)f_u(z,1)).$$

To solve this functional equation we will again employ the guess-and-check methodology.

[2]This is similar to a class of objects counted in [BEVAN, D., LEVIN, D., NUGENT, P., PANTONE, J., PUDWELL, L., RIEHL, M., AND TLACHAC, M. L. Pattern avoidance in forests of binary shrubs. arXiv:1510.08036 [math.CO], 2015].