Dyck paths and specifications

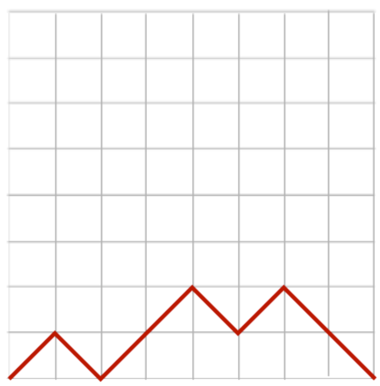
Contents

1 Dyck Paths

1.1 Decomposing Dyck paths

Definition. A Dyck path is a path on $\mathbb{Z}^2$ from $(0,0)$ to $(n,0)$ that never steps below the line $y=0$ with steps from the set $\{(1,1),(1,-1)\}$. The weight of a Dyck path is the total number of steps.

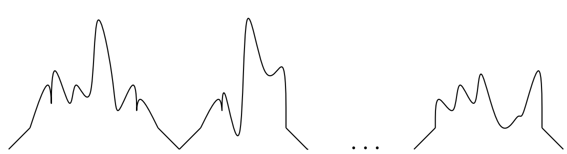
Here is a Dyck path of length 8:



Let $\mathcal{D}$ be the combinatorial class of Dyck paths.

Note that every nonempty Dyck path must begin with a $(1,1)$-step and must end with a $(1,-1)$-step.

There are a few ways to decompose Dyck paths. One way is to break it into blocks beginning with $(1,1)$, ending with $(1,-1)$ and never touching the x-axis strictly inside a block



Every Dyck path is a sequence of such blocks, so

$$\mathcal{D} = \textsc{Seq}(\mathcal{Z}_{\nearrow} \times \mathcal{D} \times \mathcal{Z}_{\searrow})$$

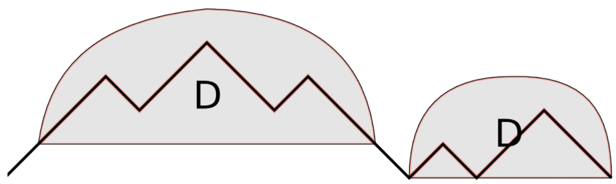
Here's another decomposition, which is a little more subtle,

Theorem. *The combinatorial class of Dyck paths denoted $\mathcal{D}$, satisfies the following combinatorial equation:*

$$\mathcal{D} = \epsilon + (\mathcal{Z}_{\nearrow} \times \mathcal{D} \times \mathcal{Z}_{\searrow} \times \mathcal{D})$$

The intuition As with the binary words, we need a distinct decomposition. I am going to consider the first time the Dyck path returns to the origin. This is a unique point in the path.

What is going on? The path first takes a step up, and just before it returns to the origin it takes a down step. Inbetween it is always above the line $y = 1$. In fact, this part inbetween it itself a Dyck path, just elevated one spot up. After we come back down, the remaining segment is also a path using the same steps which starts and ends on the axis– it too is a Dyck Path. Here is a diagram:



How to write it We prove the above theorem with a decomposition on the point of first return to the axis. Consider a genertic walk of length n. First we note that if the walk is empty, then it is a walk of size 0, and is representable by ϵ.

To end on the axis the walk must take as many up steps as downsteps. Furthermore the first step is an up step. Hence, there is some later down step. We pick the *first* time it touches the axis. This is a uniquely identifiable downstep. Say it ends at position $(k, 0)$, such that $1 \leq k \leq n$.

It may be that this is the next step, in which case there is an empty Dyck path between them. Otherwise, there is some set of steps that starts at $(1, 1)$ and ends at $(1, k-1)$. This set of steps never goes below $y = 1$, otherwise there is an earlier point that touches the axis. This is thus a Dyck path of smaller length simply shifted on the plane! Likewise, after the down step ending at $(k, 0)$, there is some (possibly empty) set of up down steps starting at the axis and ending at the axis and never going below. This gives rise to the decomposition listed in the theorem.

1.2 Connection to trees

The first decomposition gives the generating function equation

$$D(x) = \frac{1}{1 - x^2 D(x)}$$

Rearranging this is

$$D(x)(1 - x^2 D(x)) = 1 \quad \text{so} \quad x^2 D(x)^2 - D(x) + 1 = 0$$

The second decomposition gives the generating function equation

$$D(x) = 1 + x^2 D(x)^2$$

which is the same equation, although we obtained it in two very different ways.

Next solve the generating function equation

$$D(x) = \frac{1 \pm \sqrt{1 - 4x^2}}{2x^2}$$

as for binary rooted trees (seen in class)[1] the negative sign in the $\pm$ will cancel the 1s in the numerator and hence let us cancel the x^2 from the denominator. In fact matters are even closer to rooted trees: Let $t = x^2$ then the generating function becomes

$$\frac{1 - \sqrt{1 - 4t}}{2t}$$

[1]These are the kind of binary rooted tree where each vertex has a left and a right child, either or both of which can be empty.

which is exactly the generating function for binary rooted trees. This tells us a few things.

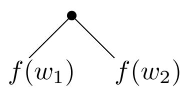
First since $D(x)$ depends only on x^2, all powers of x appearing in the generating function must be even, and so there can only be Dyck paths of even length – looking back at the definition of Dyck paths can you see why this must be true?.

Second since the generating functions are the same (after the change of variables) we know that there are the same number of Dyck paths of length $2n$ as there are binary rooted trees with n vertices. We might ask whether there is a nice way to see this second fact directly from the combinatorial objects, that is, is there a nice bijection between Dyck paths and rooted trees. Here is one such bijection.

Let w be a Dyck path. Define the tree $f(w)$ inductively as follows. If w is empty then $f(w)$ is the empty tree. Otherwise, decompose

$$w = \nearrow w_1 \searrow w_2$$

as in the second decomposition. w_1 and w_2 have strictly smaller length than w, so define

$f(w) =$ 

Now we just need to check that f is a bijection. We could do this in the usual way: check it is one-to-one and onto, but here is another approach which is often useful for these combinatorial bijections: define what should be the inverse of f, namely given a binary rooted tree T let

$$g(T) = \begin{cases} \epsilon & \text{if } T = \epsilon \\ \nearrow g(L) \searrow g(R) & \text{if } T \text{ is a root with left child } L \text{ and right child } R \end{cases}$$

Then it suffices to show $g(f(w)) = w$ for all Dyck paths w and $f(g(T)) = T$ for all trees T, and that f preserves sizes. The set-theoretic proposition we're using (applied to $\mathcal{D}_{2n}$ and $\mathcal{T}_n$) is

Proposition. *Let A and B be sets. Let $f : A \to B$ and $g : B \to A$. If $g(f(a)) = a$ for all $a \in A$ and $f(g(b)) = b$ for all $b \in B$ then f and g are both bijections.*

Proof. By symmetry it suffices to prove that f is a bijection.

Suppose $f(a_1) = f(a_2)$ with $a_1, a_2 \in A$. Then $a_1 = g(f(a_1)) = g(f(a_2)) = a_2$ so f is one-to-one.

Take $b \in B$. Then $f(g(b)) = b$ and $g(b) \in A$, so f is onto. Thus f is a bijection. □

Exercise. For $f : \mathcal{D} \to \mathcal{T}$ and $g : \mathcal{T} \to \mathcal{D}$ as defined before the proposition, prove that $f(\mathcal{D}_{2n}) \subseteq \mathcal{T}_n$ and that $g(f(w)) = w$ for all Dyck paths w and $f(g(T)) = T$ for all trees T

2 Combinatorial specifications

2.1 Recall

Elemental classes Let $\mathcal{E}$ denote the neutral class that consists of a single object of size zero. $E(z) = 1$. Let $\mathcal{Z}$ denote the atomic class that consists of a single object of size 1 called an atom. $Z(z) = z$.

Cartesian Product The cartesian product construction applied to two classes $\mathcal{B}, \mathcal{C}$ forms ordered pairs, $\mathcal{A} = \mathcal{B} \times \mathcal{C}$ iff $\mathcal{A} = \{\alpha = (\beta, \gamma) | \beta \in \mathcal{B}, \gamma \in \mathcal{C}\}$ where the size of a pair is additive, namely $|\alpha|_{\mathcal{A}} = |\beta|_{\mathcal{B}} + |\gamma|_{\mathcal{C}}$. By considering all possible pairs we see that the counting sequences satisfy $A_n = \sum_{k=0}^{n} B_k C_{n-k}$ which is exactly the product of the generating functions $A(z) = B(z) \cdot C(z)$.

Union Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be combinatorial classes such that $A = B \cup C$ and $B \cap C = \varnothing$ with size defined in a compatible way; if $\omega \in A$ then

$$|\omega|_{\mathcal{A}} = \begin{cases} |\omega|_{\mathcal{B}} & \omega \in \mathcal{B} \\ |\omega|_{\mathcal{C}} & \omega \in \mathcal{C} \end{cases}.$$

Then we clearly have $A_n = B_n + C_n$ and hence $A(z) = B(z) + C(z)$. Hence the union of disjoint sets is admissible and it translates as a sum of ogf.

Sequence If $\mathcal{B}$ is a class then the sequence class $\textsc{Seq}(\mathcal{B})$ is defined to be the infinite sum

$$\mathcal{A} = \textsc{Seq}(\mathcal{B}) = \mathcal{E} + \mathcal{B} + (\mathcal{B} \times \mathcal{B}) + (\mathcal{B} \times \mathcal{B} \times \mathcal{B}) + \dots$$

equivalently, $\mathcal{A} = \{(\beta_1, \beta_2, \dots, \beta_l) \text{ s.t. } \beta_j \in \mathcal{B}, l \geq 0\}$ This only works if $\mathcal{B}$ does not contain an element of size zero (a neutral element). Also $\alpha = (\beta_1, \beta_2, \dots, \beta_l) \Rightarrow |\alpha| = |\beta_1| + \dots + |\beta_l|$

Notice that if we want a sequence that contains exactly k-objects or at least k objects then we might write $\textsc{Seq}_k(\mathcal{B}) = \mathcal{B}^k$ and $\textsc{Seq}_{\geq k}(\mathcal{B}) = \mathcal{B}^k \times \textsc{Seq}(\mathcal{B})$

2.2 Defining a class: A specification

Definition. A specification for an r-tuple of classes $\mathcal{A} = (\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)})$ is a set of r equations

$$\begin{aligned} \mathcal{A}^{(1)} &= \Phi_1(\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)}) \\ &\vdots \\ \mathcal{A}^{(r)} &= \Phi_r(\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)}) \end{aligned}$$

where each Φ_i is built using the admisible constructions we know as well as the neutral class $\mathcal{E}$ and atomic class $\mathcal{Z}$.

You can think of drawing a graph of dependencies between these classes. If the class is acyclic then the problem can be solved iteratively. If it contains a cycle then the construction is recursive and can hopefully be solved as we solved the tree problem.

Definition. A class is said to be constructible (or specifiable) iff it admits a specification in terms of admissible operators.

We focus recursive and iterative constructions. Consider a graph of dependencies. If it is acyclic, the specification is said to be iterative. If there is a loop containing $\mathcal{A}$, then we say that $\mathcal{A}$ is recursive.

In class we've considered classes which were word families. These looked like $\mathcal{A} = \Phi(\mathcal{Z}_1, \mathcal{Z}_0)$. These are all iterative constructions. We've also considered classes of rooted trees which are recursive constructions.