LECTURE 3 – THE HIERARCHY OF GENERATING FUNCTIONS

JAY PANTONE

WILFIAN FORMULAS

In Lecture 1, we discussed four types of answers to the question of "How many objects of size n are there?" In his influential paper *What is an answer?*, Wilf presented another type of answer that has come to be known as a *Wilfian formula*.

Definition: A *Wilfian formula* for the sequence a_n is a polynomial-time algorithm in n to compute n.

In cases where the sequence $(a_n)_{n\in\mathbb{N}}$ grows exponentially, this represents a big improvement over just generating and counting all the objects. So many sequences in enumerative combinatorics don't have a closed form or even a nice recurrence that discovery of a Wilfian formula for a sequence is often considered a very good answer.

Example: We'll see a combinatorial class later with a generating function $T(z)$ that satisfies the functional equation

$$T(z) = z + T(z^2 + z^3).$$

This generating function has no (known) nice expression as a function, and the coefficients have messy asymptotic behavior:

$$a_n \sim \frac{\phi^n}{n} u(\log(n)),$$

where $\phi \approx 1.618$ is the golden ratio and $u(\cdots)$ is a positive, nonconstant, continuous function that is periodic with period $\log(4 - \pi)$. Despite this, we can use a computer to get any particular coefficient a_n in polynomial-time. Let's find a_{100}:

Maple

```
> T := z;
                              T := z

> T := expand(z + subs(z=z^2+z^3, T));
                                3    2
                          T := z  + z  + z

> T := expand(z + subs(z=z^2+z^3, T));
                   9      8      7      6      5    4    3    2
             T := z  + 3 z  + 3 z  + 2 z  + 2 z  + z  + z  + z  + z

> T := expand(z + subs(z=z^2+z^3, T)): coeff(T,z,100);
                                0

> T := expand(z + subs(z=z^2+z^3, T)): coeff(T,z,100);
                                0

> T := expand(z + subs(z=z^2+z^3, T)): coeff(T,z,100);
                         4828410152228250886

> T := expand(z + subs(z=z^2+z^3, T)): coeff(T,z,100);
                         5520498313790316062

> T := expand(z + subs(z=z^2+z^3, T)): coeff(T,z,100);
                         5520498313790316062
```

Why is this polynomial-time in n? Each substitution step consists of replacing z by $z^2 + z^3$ and expanding the (finite) polynomial. This is done quickly with the binomial theorem. Everything else is simple arithmetic.

Most of the answers given in the first lecture are themselves Wilfian formulas. Any (reasonable) closed-form is a Wilfian formula, so long as the computation itself can be done in polynomial-time. For example, the closed form of the Catalan numbers

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

constitutes a Wilfian formula.[1]

Recurrences, too, double as Wilfian formula. In the case of a linear recurrence with constant coefficients (e.g., $a_{n+1} = a_n + a_{n-1}$), we have in fact a linear time recurrence. Recurrences tend to be the fastest way to compute a_n for simple sequences. The generating function $T(z)$ is known to have coefficients that satisfy the recurrence

$$a_n = \sum_{2k+3m=n} \binom{k+m}{k} a_{k+m},$$

and this can also be used to compute a particular a_n in polynomial time.[2]

[1] While we won't get too technical on the computational complexity of the basic arithmetic operations, it suffices to know that addition, subtraction, multiplication, and division of two n-digit numbers can be performed in $O(n^2)$ time or better, and the square root of an n digit-number can be computed in $O(n^2)$ time.

[2] Take note of how a relatively simple closed form leads to very complex asymptotic behavior!

Lastly, generating functions of several forms can act as Wilfian formulas. The next section explores different types of generating functions and describes how their coefficients may be determined.

The Hierarchy of Generating Functions

We categorize generating functions by their simplicity.

- **Rational:** A generating function is *rational* if it has the form
$$f(z) = \frac{p(z)}{q(z)},$$
where $p(z)$ and $q(z)$ are polynomials (with rational coefficients). Equivalently, $f(z)$ is rational if there exist polynomials $p(z)$ and $q(z)$ such that
$$0 = q(z)f - p(z).$$

- **Algebraic:** A generating function is *algebraic* of degree m if there exist polynomials $p_0(z), p_1(z), \ldots, p_m(z)$ such that
$$0 = p_m(z)f^m + p_{m-1}(z)f^{m-1} + \cdots + p_1(z)f + p_0(z).$$
The expression on the right-hand side is called the *minimal polynomial* of the generating function. Phrase another way, $f(z)$ is algebraic if there exists a polynomial $P(z, Y) \in \mathbb{Q}[z, Y]$ such that $P(z, f) = 0$.

  Example: The generating function $f(z) = \dfrac{1 - \sqrt{1 - 4z}}{2z}$ is rational because it satisfies the equation
$$0 = zf^2 - f + 1.$$
(Proof by quadratic equation. Note that there are two "solutions" to the equation, but only one is a formal power series.)

  It's important to see that expressing generating functions as combinations of addition, multiplication, division, and roots *does not* capture all algebraic functions. So, in order to claim you have a procedure to do something to algebraic generating functions, it must operate on the level of minimal polynomials.

  Example: Suppose you're searching for the generating function $h(z)$, and you know it's the product of two generating functions $f(z)$ and $g(z)$. You also know that $f(z)$ has minimal polynomial
$$0 = f^3 + (2z + 1)f^2 + 5$$
and $g(z)$ has minimal polynomial
$$0 = (z^3 + z)g^2 + 2zg + 1.$$
It's clear that $h(z)$ should be algebraic, and of degree at most 6, but it's not obvious how to find its minimal polynomial. We'll learn how to do this later using *resultant methods*.

- **D-finite / Holonomic:** A generating function is *differentially finite* (*D-finite*) or *holonomic* if it is the solution of a linear differential equation with polynomial equations. Equivalently, $f(z)$ is D-finite of order k if there exist $p_{-1}(z), p_0(z), \ldots, p_k(z)$ such that
$$0 = p_k(z)f^{(k)} + p_{k-1}(z)f^{(k-1)} + \cdots + p_1(z)f' + p_0(z)f + p_{-1}(z).$$
As in the algebraic case, it is often impossible to express such generating functions in a closed form.

  Example: The generating function $f(z) = \sum_{n=0}^{\infty} n!x^n$ is D-finite, as it satisifes
$$z^2 f'(z) + (z-1)f(z) + 1 = 0.$$
The class of D-finite function possesses many nice closure properties, and they are easily to work with computationally.
- **D-algebraic:** A generating function $f(z)$ is *differentially algebraic* (*D-algebraic*) if there is a polynomial $P(z, Y_0, Y_1, \ldots, Y_k)$ such that
$$0 = P(z, f, f', f'', \ldots, f^{(k)}).$$
This kind of differential equation is sometimes called an *algebraic differential equation*.

  Example: The generating function $f(z) = \sec(z)$ is D-algebraic, as it satisfies the equation
$$0 = f(z)f''(z) - 2(f'(z))^2 - f(z)^2.$$

We will briefly describe properties of several of these classes. They will be discussed in more detail in future lectures.

- **Rational:**
  - Wilfian formula (proved below)
  - Coefficient extraction can be done (closed form obtainable)
$$a_n = \sum_{i=1}^{k} q_i(n)\mu_i^n,$$
where $q_i(n)$ are polynomials and μ_i are algebraic numbers.
  - Linear recurrence with constant coefficients
  - Asymptotic behavior can be fully computed:
$$a_n \sim Cn^{\alpha}\mu^n,$$
for $\alpha \in \mathbb{N}$, μ algebraic, C algebraic.
- **Algebraic:**
  - Wilfian formula
  - Given an explicit generating function, coefficient extraction can be done

  - Linear recurrence with polynomial coefficients
  - Asymptotic behavior can be fully computed:
    $$a_n \sim C n^{p/q} \mu^n,$$
    for $p/q \in \mathbb{Q}$, μ algebraic, C messy.
- **D-finite:**
  - Wilfian formula
  - Coefficient extraction usually cannot be done
  - Linear recurrence with polynomial coefficients
  - Asymptotic behavior hard (and in some doubt), constants possibly undecidable, growth rates still algebraic
- **D-algebraic:**
  - Basically nothing is known.
  - Infinitely many singularities, natural boundaries, etc
  - Exercise: Wilfian formula?

Let's prove now that rational generating functions are Wilfian formulas, without using coefficient extraction or conversion to linear recurrence. Let
$$f(z) = \frac{p(z)}{q(z)},$$
for polynomials $p(z)$ and $q(z)$ of degrees k and ℓ, respectively. Fix n. We want to determine f_n. Rearrange the equation to get
$$f(z)q(z) - p(z) = 0.$$
Now,
$$(f_0 + f_1 z + \cdots + f_n z^n + \cdots)(q_0 + q_1 z + \cdots q_\ell z^\ell) - (p_0 + p_1 z + \cdots + p_k z^l) = 0,$$
and so we find a system of equations
$$\begin{aligned} f_0 q_0 - p_0 &= 0 \\ f_1 q_0 + f_0 q_1 - p_1 &= 0 \\ &\cdots \end{aligned}$$
that can be solved line by line to get f_n. (Solving a linear system can be done in $O(n^3)$ time, plus some time to build the system.)