Formal derivatives

For a f.p.s. \( f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots \), we define

\[
f'(x) = a_1 + 2a_2 x + 3a_3 x^2 + \ldots + \ldots.
\]

Familiar rules like \((fg)' = fg' + f'g\) and the chain rule \((f(g))' = f'(g) \cdot g'\), the latter when \(g(0) = 0\), can be verified formally. Also, for any f.p.s. \( f \)

\[
a_n = \frac{1}{n!} f^{(n)}(0).
\]

Power series seen in calculus can be introduced as f.p.s. For example,

\[
\ell(x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \ldots.
\]

Check that \(\ell'(x) = 1/(1 - x)\). In calculus, \(\ell(x)\) is \(-\log(1 - x)\). It is often convenient to use this notation: If \( f \) is a f.p.s with \( f(0) = 1 \),

\[
\log(f) = -\ell(f - 1) = -(f - 1) - \frac{(f - 1)^2}{2} - \frac{(f - 1)^3}{3} - \ldots.
\]

Then \((\log f)' = f'/f\).

The QUICKSORT algorithm puts a sequence of items into linear order. For example, perhaps a list of students is to be arranged by exam scores. If there is one student, there is nothing to do. Otherwise, the first step in QUICKSORTing \( n \) students is to compare the score of one student “Bob” with the scores of the other \( n - 1 \) students, and partition the students into two sets: those with better scores than Bob and those with less good scores than Bob. Then one sorts each of these subsets recursively.

For example, to sort \( \ell, m, o, a, t, d, p \) alphabetically, we make 7 comparisons and find that \( \ell \) should be in the third position:

\[
\{a, d\} \ell \{m, o, t, p\}.
\]

One comparison is required for \( a, d \), only to discover that they are in the correct order already. Then 3 more comparisons to realize that \( m \) is before \( o, t, p \). Two more comparisons to decide that \( o \) is before \( t, p \), and one final comparison to see that \( t, p \) needs to be switched to \( p, t \). That makes 14 comparisons. We would need 21 comparisons to QUICKSORT \( a, b, c, d, e, f, g \), only to discover that they are already in the correct order.

Let \( a_n \) denote the average number of comparisons the QUICKSORT algorithm requires to sort \( n \) distinct items. For example, \( a_3 = 8/3 \). In general, we have

\[
a_n = n - 1 + \frac{1}{n} \sum_{i=1}^{n} (a_{i-1} + a_{n-i})
\]
Let \( f \) be the generating functions of \( a_0, a_1, a_2, \ldots \) (we take \( a_0 = 0 \)). Multiply the recursion by \( n \) to get rid of the fraction:

\[
na_n = n(n-1) + 2 \sum_{j=0}^{n-1} a_j.
\]

We recognize \( na_n \) as the coefficient of \( x^{n-1} \) in \( f' \); multiply the above by \( x^{n-1} \) and sum over \( n = 1, 2, 3, \ldots \) to get

\[
f' = \sum_{n=1}^\infty na_n x^{n-1} = x \sum_{n=2}^\infty n(n-1)x^{n-2} + 2 \sum_{j=0}^\infty \left( \sum_{n=j+1}^\infty x^{n-1} \right).
\]

Then the above can be written (details omitted)

\[
f' = 2x(1-x)^{-3} + 2(1-x)^{-1} f \quad \text{or} \quad f' - 2(1-x)^{-1} f = 2x(1-x)^{-3}. \tag{*}
\]

This is a first order linear differential equation.

We multiply \((*)\) by \((1-x)^2\) to get

\[
(1-x)^2 f' - 2(1-x)f = 2x(1-x)^{-1} = \frac{2}{1-x} - 2,
\]

because now the LHS has the form \((uf)' = uf' + u'f\) where \(u = (1-x)^2\); that is, \((1-x)^2 f\) is an antiderivative of the RHS. We conclude

\[
(1-x)^2 f = -2 \log(1-x) - 2x
\]

(there is no constant because \(a_0 = 0\)), and finally

\[
f = -2(x + \log(1-x))(1-x)^2 = 2 \left( \sum_{k=2}^\infty \frac{x^k}{k} \right) \left( \sum_{j=0}^\infty (j+1)x^j \right).
\]

Comparison of the coefficient of \( x^n \) on both sides of the above equation gives

\[
a_n = 2 \sum_{k=2}^n \frac{n-k+1}{k} = 2(n+1) \sum_{k=1}^n \frac{1}{k} - 4n \\
\approx 2(n+1)(\gamma + \log n) - 4n = 2n \log n + O(n).
\]

Here \(\gamma\) is Euler’s constant 0.5772....
Infinite products

We can define the product of infinitely many f.p.s. \( f_i(x) \) when all but finitely many of the power series have constant term 1, and the coefficient of \( x^k, k > 0 \), is zero in all but finitely many of the factors. This means that the evaluation of the coefficient of any \( x^n \) is a finite procedure.

A famous example is the generating function of partitions of integers. A partition of an integer \( n \) is a multiset of positive integers that sum to \( n \). They may also be thought of as a solutions of \( n = i_1 + 2i_2 + 3i_3 + 4i_4 + \ldots \) in nonnegative integers \( i_j \); here \( i_j \) is the number of parts of size \( j, j = 1, 2, 3, \ldots \) (The “parts” are positive integers but the numbers of parts of various sizes are nonnegative intgers.) Let \( p(n) \) be the number of partitions of \( n \) and \( P(x) = \sum_n p(n)x^n \). Then

\[
P(x) = (1 + x + x^2 + x^3 + \ldots)(1 + x^2 + x^4 + x^6 + \ldots)(1 + x^3 + x^6 + x^9 + \ldots) \cdots = \prod_{k=1}^{\infty} (1-x^k)^{-1}.
\]

For example, \( p(5) = 7 \):

\[5 = 5, \quad 5 = 4+1, \quad 5 = 3+2, \quad 5 = 3+1+1, \quad 5 = 2+2+1, \quad 5 = 2+1+1+1, \quad 5 = 1+1+1+1+1.\]

We remark that \( p(n) \) is, very roughly, about \( 10^{\sqrt{n}} \); search the web for better approximations (look for “partition number theory”). There is a huge literature on partitions and varieties of partitions.

**Theorem.** The number of partitions of a number \( n \) into distinct parts is the same as the number of partitions of \( n \) into odd parts.

Example: Using a condensed notation, the partitions of 7 into distinct parts are 7, 61, 52, 43, 421, and the partitions of 7 into odd parts are 7, 511, 331, 31111, 1111111.

**Proof:** The generating function of the numbers of partitions into distinct parts is

\[
P_D(x) = (1 + x)(1 + x^2)(1 + x^3)(1 + x^4) \cdots
\]

and the generating function of the numbers of partitions into distinct parts is

\[
P_O(x) = (1 - x)^{-1}(1 - x^3)^{-1}(1 - x^5)^{-1}(1 - x^7)^{-1} \cdots.
\]

We may use \( 1 + x^k = (1 - x^{2k})/(1 - x^k) \) to write

\[
P_D(x) = \frac{(1 - x^2)(1 - x^4)(1 - x^6)(1 - x^8)(1 - x^{10})(1 - x^{12})(1 - x^{14})(1 - x^{16}) \cdots}{(1 - x)(1 - x^2)(1 - x^3)(1 - x^4)(1 - x^5)(1 - x^6)(1 - x^7)(1 - x^8) \cdots}.
\]

We may cancel the terms \( (1 - x^2), (1 - x^4), (1 - x^6) \), etc., from both the numerator and denominator of this rational function, and we get exactly \( P_O(x) \). \( \square \)
In case the reader is not completely sure about this infinite cancellation, remember that the coefficient of $x^{100}$, for example, in the large fraction is the same as the coefficient of $x^{100}$ in

$$\frac{(1 - x^2)(1 - x^4)(1 - x^6) \cdots (1 - x^{98})(1 - x^{100})}{(1 - x)(1 - x^2)(1 - x^3)(1 - x^4)(1 - x^5)(1 - x^6)(1 - x^7) \cdots (1 - x^{99})(1 - x^{100})},$$

since the other terms contribute nothing to this coefficient. And everyone agrees that we can cancel in the numerator and denominator of this rational function to get

$$\frac{1}{(1 - x)(1 - x^3)(1 - x^5)(1 - x^7) \cdots (1 - x^{99})}.$$