Euler’s recursion for partitions

Leonhard Euler did many extensive computations and so discovered many things. He did not have what we would regard as a proof, in many cases. An example is his recurrence relation for the numbers $p(n)$. We know

$$P(x) = \sum_n p(n)x^n = \frac{1}{(1-x)(1-x^2)(1-x^3)(1-x^4)\cdots}.$$  

Euler considered the denominator and calculated many coefficients. With the aid of Mathematica, we find the first 101 coefficients of $\prod_n (1 - x^n)$ to be

$$1 - x - x^2 + x^5 + x^7 - x^{12} - x^{15} + x^{22} + x^{26} - x^{35} - x^{40} + x^{51} + x^{57} - x^{70} - x^{77} + x^{92} + x^{100} - \ldots \quad (\ast)$$

The coefficients appear to be 1, $-1$, or 0 (mostly 0’s). Euler guessed the rule

$$\prod_n (1 - x^n) = 1 + \sum_{k=1}^{\infty} (-1)^k \left( x^{k(3k-1)/2} + x^{k(3k+1)/2} \right).$$

Proving this identity is NOT easy, and we will not include one in this course.

When we multiply $\sum_n p(n)x^n$ by the f.p.s. in $(\ast)$, we get 1, and so a recursion: For $n > 0$,

$$p(n) = p(n-1) + p(n-2) - p(n-5) - p(n-7) + p(n-12) + p(n-15) - \ldots,$$

where we understand $p(k) = 0$ for all negative $k$. The sequence $p(n)$ starts off like the Fibonacci numbers but soon diverges.

$$p(0), p(1), p(2), \ldots = 1, 1, 2, 3, 5, 7, 11, 15, \ldots.$$  

Decompositions of sets of binary strings

We will speak of binary strings of $a$’s and $b$’s in this section, rather than of 0’s and 1’s.

If $S$ and $A, B, C, \ldots, Z$ are sets of binary strings, we write

$$S = ABC \cdots Z$$

and call this expression a decomposition of $S$ when each string $s$ can be written uniquely as a concatenation of strings $a, b, c, \ldots, z$ in the respective sets. An example of a decomposition is

$$\{\epsilon, a, bab, ba, aba, babba, bba, abba, babbaa\} = \{\epsilon, a, bab\} \{\epsilon, ba, bba\}.$$
Here $\epsilon$ denotes the empty string.

We use the notation $A^*$ for the set of strings that are concatenations of any number of strings from $A$, but we ONLY use this notation when we get a set, i.e. when no string arises more than once in this way. That is, $A^*$ is the union of $\{\epsilon\}, A, AA, AAA, \ldots$ when these products are decompositions and are disjoint.

For example, let $F$ be the set of Fibonacci strings, the binary strings with no two consecutive $b$'s. Let

$$A = \{a\}, \quad B = \{ba, baa, baaa, baaaa, \ldots\}, \quad C = \{\epsilon, b\}.$$  

Then

$$F = A^* B^* C$$

is a decomposition of $F$. To repeat, this means that any Fibonacci string can be written uniquely as the concatenation of three strings, one from $A^*$ (itself uniquely a concatenation of strings from $A$), followed by one from $B^*$, and then followed by one from $C$. For example, the Fibonacci string

$$aabaababaaabab = (aa)(baaabaaaba)(b)$$

is the concatenation of $aa \in AA$, $baaabaaaba \in BBBB$, and $b \in C$. By the way, this particular decomposition of the Fibonacci strings is needlessly complex. We also have

$$F = D E^*$$

where $D = \{\epsilon, b\}$ and $E = \{a, ab\}$.

Given a set $S$ of strings, finite or infinite, we introduce the generating function

$$S(x) = \sum_{s \in S} x^{\text{length}(s)} = \sum_{n=0}^{\infty} \ell_S(n) x^n$$

where $\ell_S(n)$ is the number of strings $s \in S$ of length $n$. In our example, $A(x) = 1 + x + x^3$ and $B(x) = 1 + x^2 + x^3$.

Be careful: in this section we are using captial letters like $A$ for sets while $A(x)$ is a the f.p.s. related to $A$ as just defined.

**Proposition.** If $D = BC$ is a decomposition, then

$$D(x) = B(x)C(x).$$

Also,

$$A^*(x) = 1 + A(x) + (AA)(x) + (AAA)(x) + \ldots = (1 - A(x))^{-1}.$$  

**Proof:** A string of length $n$ is the unique concatenation of a string in $B$, say of length $i$, and a string in $C$ of length $n - i$. That is, the number $\ell_D(n)$ of strings of length $n$ in $D$
is \( \sum_{i=0}^{n} \ell_B(i) \ell_C(n-i) \). This proves the first assertion. The second assertion is now clear, because e.g. \((AAA)(x) = (A(x))^3\) by the first assertion.

The generating functions of \( A, B, C \) by length are, respectively, \( A(x) = x, \quad B(x) = x^2(1 - x), \quad C(x) = 1 + x \). Thus the generating function \( F(x) = \sum_n f_n x^n \) of the Fibonacci strings by length is

\[
F(x) = (1 - x)^{-1}(1 - x^2(1 - x)^{-1})^{-1}(1 + x) = \frac{1 + x}{1 - x - x^2}.
\]

Of course, it would be quicker to use our second decomposition of \( F \) as \( \{\epsilon, b\} \{a, ab\}^\ast \) because then we immediately have

\[
F(x) = (1 + x) \frac{1}{1 - (x + x^2)}.
\]

As an example, consider binary strings where an odd block of \( a \)'s is never followed by an odd block of \( b \)'s. The set \( S \) of all such strings can be decomposed as \( \{b\}^\ast M \{a\}^\ast \) where

\[
M = \{a\} \{a\}^\ast \{b\} \{b\}^\ast \setminus \{a\} \{aa\}^\ast \{b\} \{bb\}^\ast.
\]

(The first term represents all strings of at least one \( a \) followed by a string of at least one \( b \) while the second is the set of strings consisting of an odd number of \( a \)'s followed by a string of an odd number of \( b \)'s.

We find

\[
M(x) = \frac{x^3(2 + x)}{(1 - x)^2}
\]

and then

\[
S(x) = \frac{(1 + x)^2}{1 - 2x^2 + 2x^3}.
\]

This means, in particular, that the numbers \( s_n \) of such strings satisfy the linear recurrence

\[
s_n = 2s_{n-2} - 2s_{n-3}.
\]

This would not be clear without the decomposition method.

The “decomposition” idea makes other things simple too. We can quickly write down the generating function by length of the binary strings with no three consecutive \( b \)'s. Or where there are no three consecutive \( b \)'s and \( a \)'s occur only in consecutive ‘blocks’ of odd lengths. Or we could consider ternary strings (of \( a \)'s, \( b \)'s, and \( c \)'s) with no two consecutive \( c \)'s, etc.