COUNTING IN A SOPHISTICATED MANNER

ALGEBRAIC COMBINATORICS (MATH 68), BY NADIA LAFRENIÈRE

ABSTRACT. We often hear that combinatorialists count things in original ways, but what does that mean? This lecture is an introduction to (enumerative) combinatorics.

1. What do we count?

Usually, we are interested in (finite or countable) families of finite sets. Each set should be indexed by an integer, for example.

Example.

- The subsets of $[n] = \{1, 2, \ldots, n\}$.
- The permutations of $[n]$.
- The compositions of an integer n, i.e. tuples $(\alpha_1, \ldots, \alpha_k)$ such that $\alpha_1 + \ldots + \alpha_k = n$ and all the α_i's are positive integers.

We are interested in the counting function of a family that, to each S_i, $i \in \mathbb{N}$, gives its number of items:

$$\begin{aligned} f : \mathbb{N} &\to \mathbb{N} \\ i &\mapsto \#S_i. \end{aligned}$$

Doing this process, we count simultaneously the number of items in all the sets.

Remark. Doing this way also allows us to refine the answer to unanswerable questions. For example, if I'm asking "How many sets of integers have cardinality n?", then the answer is of course infinitely many. However, refining the question, I could ask the number of subsets of $[m]$ that have n elements, and the answer would be $\binom{m}{n}$. By the way, $\sum_{m=0}^{\infty} \binom{m}{n} = +\infty$.

2. How to count?

There are many standard ways to express the counting function. In his book on Enumerative Combinatorics [EC1], Richard P. Stanley presents five of them.

2.1. A close formula.

This is our golden standard for representing counting functions. They are often explicit and easy to compute. We know many common examples:

- $\#\mathcal{S}_n = n!$ (the permutations of $[n]$).

- $\#\mathcal{P}([n]) = 2^n$ (subsets of $[n]$).
- $\#\{\text{Subsets of } [n] \text{ of size } k\} = \binom{n}{k}$.
- The number of compositions of n is 2^{n-1}. (To get to that result, draw n dots on a line, and look at the set of spaces between two dots (there are $n-1$ such spaces. Each subset of these spaces is in bijection with a composition.)

  Example. The 8 compositions of 4:

– (4)	• • • •	↔ {}	– (1, 3)	•| • • •	↔ {1}
– (3, 1)	• • •|•	↔ {3}	– (1, 2, 1)	•| • •|•	↔ {1, 3}
– (2, 2)	• •| • •	↔ {2}	– (1, 1, 2)	•|•| • •	↔ {1, 2}
– (2, 1, 1)	• •|•|•	↔ {2, 3}	– (1, 1, 1, 1)	•|•|•|•	↔ {1, 2, 3}

Sadly, we do not know a close formula for a lot of objects. Sometimes, we can however count it in another way.

2.2. **A recurrence.** A recurrence is also a good way to express some properties related to an object, sometimes (i.e. for some properties) even much better than a recurrence relation.

2.2.1. *Basic example: The Fibonacci numbers.* An interesting example is the terms in the Fibonacci sequence: $F_{n+1} = F_n + F_{n-1}$, with $F_0 = 0$ and $F_1 = 1$. With the sequence put in this form, it is at least obvious that the sequence is increasing, and that it contains only positive integers. The same observations are not too hard to make with the closed formula for the Fibonacci numbers: $F_n = \frac{1}{2^n\sqrt{5}} \cdot \left((1+\sqrt{5})^n - (1-\sqrt{5})^n\right)$. However, the reader must do some verification in order to get the properties.
The main problem with the closed formula for the Fibonacci numbers, at least to the eyes of a combinatorialist, is that it involves an irrational number ($\sqrt{5}$), and what is not trivial from the formula is that the result will always be an integer.
Why is the Fibonacci sequence interesting for a combinatorialist? What does it count?
The Fibonacci numbers count the number of couples of rabbits that are raised in captivity, the number of ancestors in the genealogy of drone bees (the male was an unfertilized egg), and the number of petals a sunflower has on a given row. Also, it is the number of possible rhythms a verse may have in a sanskrit poem, where each syllable last either one time or two times. (For more details on this, read the insert called "Fibonacci Rhythms" in the Quanta Magazine's paper [Kla14].)

Problem 1. *Read the text in Quanta Magazine and prove that the number of rhythms with n times is indeed the n-th Fibonacci number.*

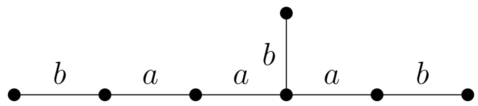

FIGURE 1. A tree with more non-empty palindromes than edges.

2.3. **An algorithm.** This way of counting includes the last two ones, but is much larger. Algorithms are particularly useful to compute some values over an example, or countably many examples, and it can also count objects that we know how to generate, but we do not know how to count.

A very good example of an algorithm to come out with an answer that is an integer is the Todd-Coxeter algorithm. Given a group G that is represented using generators and relations, it counts the number of (left) cosets $[G : H]$ for a subgroup H (if you do not know what all this means, it is outside the scope of this course, so it is fine). Another example that we will cover in this class (around the end of October) is the Robinson-Schensted algorithm to count the length of the longest increasing sequence(s) in a permutation. And this algorithm does even better: it gives an example of a longest increasing sequence.

2.4. **The asymptotic behavior.** Even with a recurrence or a closed formula, it could be hard to answer the question *"How big this family can get when n gets big?"* Then, an asymptotic approximation could even be more useful for some purposes.

2.4.1. *Binary matrices with exactly 3* 1*'s in each row and in each column.* We are looking here only at $n \times n$ matrices. It is easy to see that the counting function f satisfies $f(1) = f(2) = 0$ and $f(3) = 1$. There is a closed formula for $f(n)$:

$$f(n) = 6^{-n} n!^2 \sum_{\alpha,\beta,\gamma \in \mathbb{N},\ \alpha+\beta+\gamma=n} \frac{(-1)^\beta (\beta+3\gamma)! 2^\alpha 3^\beta}{\alpha! \beta! \gamma!^2 6^\gamma}. \tag{1}$$

Even though this may be useful for explicit computations, I personally don't see anything in it (except for a vague intuition it should be combinatorial). In counterpart, we know something about the asymptotic behaviour:

$$\lim_{n\to\infty} \frac{e^2 36^{-n}(3n)!}{f(n)} = 1,$$

which means we could accept $e^{-2} 36^{-n}(3n)!$ as a description of $f(n)$ when n is big.

Problem 2. *Prove combinatorially that formula* (1) *is right when n=4.*

2.4.2. *Palindromes in a tree.* The language of a labeled tree is the set of all the words that one can read while walking along its paths. For example, the tree in Figure 1 has language

$$\{\epsilon \text{ (the empty word)}, a, b, aa, aaa, ab, ba, aab, baa, bab, aaab, baaa, baab, baaab\}.$$

The language of palindromes of a tree is the restriction of its language only to the palindromes (words that read the same from left to right and from right to left). For example, the language of palindromes of the tree in Figure 1 is of size 8, since it contains ϵ, a, aa, aaa, b, bab, $baab$ and $baaab$. This makes it the smallest tree having more non-empty palindromes (7) than edges (6). This is interesting, because this initiated the quest to the answer to the following question: *"What is the maximal size for the language of palindromes of a tree with* n *edges"?* Define $f(n)$ to be that size. We can create infinitely many trees of the form of the comb, pictured in Figure 2, and we can verify that combs all have $cn\sqrt{n}$ palindromes, for some real number c. Moreover, we

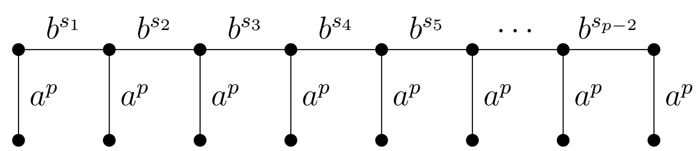

FIGURE 2. A *comb* is a tree with a main branch, called the *spine*, on which the edges are label by a different letter than the other branches. The number of edges in segments of the spine follows a Sidon set, i.e. a set $\{s_1, \ldots, s_n\}$ such that $s_i + s_j \neq s_k + s_l$ whenever $\{i, j\} \neq \{k, l\}$.

can prove that trees having a given property (that we will call *Property* X), including the combs, all have at most $Cn\sqrt{n}$ palindromes, for another real number C. Hence, this means that the restriction of f to trees satisfying property X is the function $g(n)$ such that

$$cn\sqrt{n} \leq g(n) \leq Cn\sqrt{n},$$

and that g is asymptotically of the order on $n\sqrt{n}$ (we write $g(n) \in \Theta(n\sqrt{n})$).
In fact, the function $f(n)$ satisfies the same order of magnitude (as shown in [GKRW15]), which means that the maximal number of palindromes a tree with n edges can have is at most a constant times $n\sqrt{n}$. We do not know any exact formula for this, since computing *all* the trees with every possible labeling proved way too costly in time.
(This scientific journey lead to a paper [BLP15], from which the pictures were taken.)

2.5. **A generating function.** A generating function is a way of describing symbolically a counting function:

$$\sum_{n\geq 0} f(n)x^n.$$ [1]

We then look for close forms to express these sums. Still in Enumerative combinatorics, Richard Stanley describes generating functions as *"the most useful but most difficult to understand method for evaluating"* a counting function. [EC1] The advantage lies in the fact that one can easily add functions, multiply them (through convolution), and even differentiate them, in order to establish a right counting function.
Another good thing about generating function in combinatorics, is that we consider them as *formal* power series. That means, for short, that we do not need to verify their convergence.

2.5.1. *43 nuggets at McDonald's.* In the UK, McDonald's used to sell chicken McNuggets in pack of 6, 9 and 20. The question is *"What is the maximum number of nuggets one could not buy?"* Generating functions could be great non only to solve that problem (that can also be solved using elementary arithmetic), but also to know, for possible numbers of nuggets, how many packs of each one would need. I have a friend who is interested in this, and he suggested using a computer to solve that problem, through generating function[2].
The original problem could then be rephrased as *"What is the maximum number n for which the coefficient in front of x^n is 0 in the (ordinary) generating function for nuggets?"* Here is that function :

$$F(x) = \sum_{n\geq 0}(x^6 + x^9 + x^{20})^n = \frac{1}{1 - x^6 - x^9 - x^{20}}.$$

Computing the expansion of that formula, we would have the first number. After some computations (Figure 3), we get that 43 is the maximum number of nuggets one cannot order (see this Numberphile video for some fun idea of activities to do at a drive-thru).

It is already more powerful if we want to get more details. For example, if we want to know how many nuggets we should order, we could grade the formula by using different variables for packs of 6, 9 and 20. Same thing if we want to have exactly k packs of

[1] This is the *ordinary* generating function. We will see next week that the *exponential generating function* follows the same idea, and is described as $\sum_{n\geq 0} f(n)\frac{x^n}{n!}$.

[2] I owe some thanks to Duncan Levear for showing me that problem from a generating function perspective.

A number theory problem (OGFs)

McDonald's sells chicken nuggets in packs of 6, 9, and 20. What is the largest number that cannot be made out of these?

In [1]: Slide Type Fragment

```
%display latex
S=(1/(1-x^6-x^9-x^20)).series(x,100); S
```

Out[1]:

$$1+1x^{6}+1x^{9}+1x^{12}+2x^{15}+2x^{18}+1x^{20}+3x^{21}+4x^{24}+2x^{26}+5x^{27}+2x^{29}+7x^{30}+3x^{32}+9x^{33}+6x^{35}+12x^{36}+7x^{38}+16x^{39}+1x^{40}+12x^{41}+21x^{42}+17x^{44}+28x^{45}+3x^{46}+24x^{47}+37x^{48}+3x^{49}+36x^{50}+49x^{51}+6x^{52}+50x^{53}+65x^{54}+12x^{55}+72x^{56}+86x^{57}+16x^{58}+102x^{59}+115x^{60}+30x^{61}+143x^{62}+151x^{63}+45x^{64}+202x^{65}+204x^{66}+70x^{67}+282x^{68}+269x^{69}+111x^{70}+394x^{71}+361x^{72}+165x^{73}+549x^{74}+485x^{75}+253x^{76}+762x^{77}+646x^{78}+378x^{79}+1058x^{80}+876x^{81}+561x^{82}+1462x^{83}+1176x^{84}+833x^{85}+2024x^{86}+1592x^{87}+1221x^{88}+2789x^{89}+2163x^{90}+1788x^{91}+3847x^{92}+2933x^{93}+2603x^{94}+5298x^{95}+4008x^{96}+3771x^{97}+7282x^{98}+5474x^{99}+\mathcal{O}\left(x^{100}\right)$$

In [2]: Slide Type Fragment

```
N = [str(t[1]) for t in S.coefficients() if t[0]==0]
nonuggets = ', '.join(N)
print("You cannot make packs of " + nonuggets)
```

```
You cannot make packs of 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 22, 23, 25, 28, 31, 34, 37, 43
```

FIGURE 3. Computing the coefficients of a generating function with a software is straightforward.

Slide Type Sub-Slide

Example 2

From the McDonald's example, we see that there are 115 different ways to make 60 chicken nuggets. How many of those ways use k packs?

In [5]: Slide Type Fragment

```
(1/(1-y*x^6 - y*x^9 - y*x^20)).series(x,100).coefficients()[60][0].expand()
```

Out[5]: $y^{10}+36\,y^{9}+70\,y^{8}+7\,y^{7}+y^{3}$

FIGURE 4. The power of generating functions is often to get more detailed solutions.

nuggets. Then, the generating function would look like

$$F(x)=\sum_{n\geq 0}(yx^6+yx^9+yx^{20})^n=\frac{1}{1-y(x^6-x^9-x^{20})},$$

and we would be interested only in the terms that have k as a power for the y variable. This can be seen in Figure 4.

Maybe that problem would have look more like a real-life problem if I had phrased that in terms of money and coins... That would work as well.

3. One statistics, three different ways to count.

A *derangement* is a permutation with no fixed point. Write d_n the number of derangements for permutations of n objects. One can easily see that $d_0 = d_2 = 1$, $d_1 = 0$ and $d_3 = 2$. This could be much harder as n increases! As a homework assignment, you will be guided in three different ways to compute them (Recursive formula, exponential generating function and asymptotic behavior). Unfortunately, there is no known close formula for the derangements.

4. The validity of a counting function

Now that we know many ways to express a counting function, we still need to verify that a given function is right, which means that it counts the thing it is supposed to. A nice way to prove that two objects (or sets) have the same properties is by describing an isomorphism. It is much better than only showing there exists one. Faithful to that principle, a combinatorial proof is one that shows explicitly that bijection, rather than simply proving than the two sets have the same cardinality.

Example. The following formula is true:

$$\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}. \tag{2}$$

From a non-combinatorial point of view, the following solution would look acceptable:

$$\sum_{k=0}^{n} k\binom{n}{k} = \sum_{k=0}^{n} k\frac{n!}{k!(n-k)!} = n\sum_{k=0}^{n} \frac{(n-1)!}{(k-1)!(n-k)!} = n\sum_{k=0}^{n} \binom{n-1}{k-1} = n2^{n-1}.$$

Even though we understand all the steps, it is hard to explain in a short argument why this is true. A combinatorial proof would do a much better job.

Problem 3. *Find a combinatorial proof of formula* (2)*. This is also* #3(*a*) *of the first assigment.*

4.1. **Giving a combinatorial proof without knowing the numbers.**

We can even make a valid combinatorial proof for a counting function without even knowing the actual values it takes. Usually, this takes the form of a bijection between two sets for which we do not know the values of the counting function.

4.1.1. *Partitions of n.* A partition λ of an integer n is a set of k numbers $\{\lambda_1, \dots, \lambda_k\} \vdash n$ whose sum is n. The λ_i's are called the *parts*, and there could be many occurrences of the same number in the set.

Proposition 1 (Euler[3])**.** *Let $q(n)$ be the number of partitions of n with all parts of all distinct sizes, and let $p(n)$ be the number of partitions of n with all odd parts. Then, $p(n) = q(n)$ for all $n \geq 0$.*

[3] As cited in `http://oeis.org/A000009`.

Proof. [4] Let $\lambda \vdash n$ be a partition with only odd parts. From it, we will build μ, a partition with parts of all distinct sizes. Let r_j be the number of occurrences of $2j-1$ in λ. Then, we add to μ the parts $(2j-1)2^i$ if and only if the binary expansion of r_j contains 2^i.

Problem 4. *Prove that this function is an injective map from odd partitions to partitions with all distinct parts.*

Example. Let $\lambda = \{9^5, 5^{12}, 3^2, 1^3\} \vdash 114$. Then,

$$\mu = \{4 \cdot 9, 1 \cdot 9, 8 \cdot 5, 4 \cdot 5, 2 \cdot 3, 2 \cdot 1, 1 \cdot 1\} = \{36, 9, 40, 20, 6, 2, 1\} \vdash 114.$$

Problem 5. *Find the inverse of this algorithm, and test it on μ to check that you get λ.*

□

References

[AZ10] M. Aigner and G. M. Ziegler. *Proofs from THE BOOK (4th ed.)*. Springer, 2010.

[BLP15] S. Brlek, N. Lafrenière, and X. Provençal. Palindromic complexity of trees. In *Developments in language theory*, volume 9168 of *Lecture Notes in Computer Science*, pages 155–166. Springer, Cham, 2015. Freely accessible on ArXiv at `https://arxiv.org/abs/1505.02695`.

[EC1] R. P. Stanley. *Enumerative combinatorics. Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012. Freely available online at `http://www-math.mit.edu/~rstan/ec/ec1.pdf`.

[GKRW15] P. Gawrychowski, T. Kociumaka, W. Rytter, and T. Walen. Tight bound for the number of distinct palindromes in a tree. In *String Processing and Information Retrieval - 22nd International Symposium, SPIRE 2015*, pages 270–276, 2015.

[Kla14] E. Klarreich. The musical, magical number theorist. *Quanta Magazine*, August 12, 2014. `https://www.quantamagazine.org/number-theorist-manjul-bhargava-is-awarded-fields-medal-20140812#`.

[4]This proof is not Euler's proof, since he proved it doing formal computations. This proof is due to J.W.L. Glaisher (1907), as cited in [AZ10].