MATH 239: Introduction to Combinatorics
Part I: Enumeration

Douglas Stebila

University of Waterloo, Spring 2023

Course outline

Lecture: May 8

About me: prof in C&O, I do research in cryptography

Rest of the teaching team: Ashwin Nayak, Logan Crew, Olya Mandelshtam

MATH 239 is about counting, colouring, and connecting the dots (and a lot more). It covers two areas of combinatorics: algebraic enumeration and graph theory.

An example enumeration problem is: how many ways are there to write 6 as a sum of odd numbers?

An example graph theory problem is: can we cross every bridge in this city exactly once and return to our starting point?

In MATH 239 we continue to focus on developing proof skills.

Course notes are available on LEARN for free: two sections, one for enumeration and one for graph theory.

Course outline on LEARN. We'll spend about 5 weeks on enumeration and 7 weeks on graph theory.

Primary means of communication: Office hours or Piazza. Public posts for general questions, private posts for remark requests or if you're revealing a solution, etc. Please label your questions to help make searching easier. We encourage you to try answering each others' questions, and you can earn up to 2% bonus marks for doing so. Use the "good question" and "good answer" type links to upvote good participation.

Tutorials: in person tutorials with tutors that are focused on collaboratively solving problems and communicating your solutions. They start this Friday. There are also tutorial problems on LEARN with video solutions, but I encourage you to try the problems before watching the solutions.

Office hours: details to be posted on LEARN. Starting next week. My office hours will be Mondays 3:30–4:30 in MC 5132.

Assignments: 9 assignments due Thursdays at 6pm on Crowdmark. Best 7 out of 9 count towards your final grade. No late submissions. You will be graded on quality of presentation and communication, since this is a proof-based course.

Midterm: July 6 from 4:30-6:20pm.

Final exam during exam period.

Grading scheme: 20% assignments, 30% midterm, 50% final.

Self-declared student absences: For assignments, no effect: we already drop lowest 2 out of 9, self-declared absence will not change that. For midterm, weight will be shifted to final. Self-declared absences don't apply to the final, need a verification of illness form to request an INC.

Covid guidelines:

- Stay up to date on vaccinations – boosters available on campus
- Wearing a mask is recommended. I'll be wearing mask in enclosed environments like office hours and appreciate if you do the same.
- If you are sick, stay away. Flexibility built into grading scheme to accommodate missed assignments and exams.

What to do this week?

- Log in to LEARN
- Read the course outline on LEARN
- Download the part I course notes from LEARN and start reading
- Log in to Piazza

- Start on the first video tutorial problems
- Start on the assignment after it's posted on Thusday

1 Basic Principles of Enumeration

How many ways can we

- Put 5 objects in order?
- Choose an apple or a banana from a table with 4 apples and 7 bananas?
- Make change for \$1.00?
- Bake a pizza with 4 toppings from a list of 20?
- Write a binary string of length 10 not containing '101'?

We'll develop notation to express problems like this and use combinatorics and algebra to solve them.

1.1 AND versus OR – Products versus Sums (§1.1.1)

Let $A = \{\text{apple}, \text{orange}, \text{banana}, \text{grape}\}$ and $B = \{\text{carrot}, \text{lettuce}, \text{onion}\}$.

The number of ways of choosing a fruit from A **AND** a vegetable from B is $|A| \cdot |B| = 4 \cdot 3 = 12$.

This is equivalent to choosing one element from the **Cartesian product** of A and B:

$$A \times B = \{(a, b) : a \in A \text{ and } b \in B\}$$

For finite sets, the cardinalities satisfy

$$|A \times B| = |A| \cdot |B|$$

The number of ways of choosing a fruit from A **OR** a vegetable from B is $|A| + |B| = 4 + 3 = 7$.

This is equivalent to choosing one element from the **union** of A and B:

$$A \cup B = \{x : x \in A \text{ or } x \in B\}$$

For finite **disjoint** sets, the cardinalities satisfy

$$|A \cup B| = |A| + |B|$$

Not all sets may be disjoint; consider $A = \{\text{apple}, \text{orange}, \text{banana}, \text{grape}, \text{tomato}\}$ and $B = \{\text{carrot}, \text{lettuce}, \text{onion}, \text{tomato}\}$.

The intersection is

$$A \cap B = \{x : x \in A \text{ and } x \in B\}$$

For finite sets, we have that

$$|A \cap B| + |A \cup B| = |A| + |B|$$

1.2 Lists, permutations, and subsets (§1.1.2, §1.1.3)

A **list** of a set S is a list of the elements of S exactly once, in some order. For example, $S = \{a, b, c\}$ has 6 lists:

$$abc \quad acb \quad bac \quad bca \quad cab \quad cba$$

A **permutation** is a list of the set $\{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

We can construct a list of S by choosing any element $v \in S$ to be the first element of the list, and then append a list of the set $S \setminus \{v\}$. If p_n is the number of lists of an n-element set S for $n \in \mathbb{N}$, then

$$p_n = n \cdot p_{n-1}$$

It is easy to see that $p_1 = 1$, so by induction:

Lecture: May 10

Theorem 1 (Theorem 1.2 in the notes)**.** *For every $n \geq 1$, the number of lists of an n-element set of S is*

$$n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$$

which is denoted $n!$ ("n factorial").

By convention, we take $0! = 1$.

A **subset** of S is a collection of some (perhaps none or all) of the elements of S, at most once each and in no particular order.

How many subsets of an n-element set S are there?

To specify a subset X of S, for each element of $v \in S$ we have to decide with v is in X or not in X ("out" of X). Equivalently, we have to choose an element from $\{\text{in}, \text{out}\}$ n times, for which there are

$$\underbrace{|\{\text{in}, \text{out}\}| \cdot |\{\text{in}, \text{out}\}| \cdot \ldots \cdot |\{\text{in}, \text{out}\}|}_{n} = \underbrace{2 \cdot 2 \cdot \ldots \cdot 2}_{n} = 2^n$$

For example if $S = \{a, b, c\}$ then the subset $X = \{a, b\}$ corresponds to the list $(\text{in}, \text{in}, \text{out})$.

A **partial list** of a set S is a list of a subset of S.

We are interested in length-k partial lists of a set S of size n; in other words, k-tuples $(s_1, \ldots, s_k)$ of distinct elements of S. There are n choices for s_1 and $n-1$ choices for s_2 and ... and $(n-k+1)$ choices for s_k. So:

Theorem 2 (Theorem 1.4 in the notes)**.** *For $n, k \geq 0$, the number of partial lists of length k of an n-element set is $n(n-1)\ldots(n-k+2)(n-k+1)$.*

Do I need to restrict $k \leq n$? No, it's fine if $k > n$: one of the factors will be 0, so the whole product is 0, and there are no length-k partial lists.

When $0 \leq k \leq n$, we can write

$$n(n-1)\ldots(n-k+1) = \frac{n(n-1)\ldots \cdot 1}{(n-k)(n-k-1)\ldots \cdot 1} = \frac{n!}{(n-k)!}$$

For $n, k \geq 0$, let $\binom{n}{k}$ ("n choose k") denote the number of k-element subsets of an n-element set S. It is easy to see

- $\binom{n}{0} = 1$
- $\binom{n}{n} = 1$ for all n
- $\binom{n}{1} = n$ for all n
- $\binom{n}{k} = 0$ for $k > n$

Theorem 3 (Theorem 1.5 in the notes)**.** *For $0 \leq k \leq n$, we have that*

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Proof. Let S be a set of size n. From Theorem 1.4, the number of partial lists of S of length k is $\frac{n!}{(n-k)!}$. Let's consider another way of constructing a partial list of S length k. We will choose a k-element subset X of S AND a list of X, which will give a list of a subset of S of length k.

Thus the number of length-k partial lists of S is

$$(\#\ k\text{-element subsets of } S) \cdot (\#\text{ lists of a } k\text{-element set}) = \binom{n}{k} k!$$

So

$$\frac{n!}{(n-k)!} = \binom{n}{k} k!$$

and thus

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

□

This proof has a pattern that we'll return to many times in this course: a "combinatorial proof" where we prove an identity by counting the size of the same set in two different ways.

Example 1 (Example 1.6 in the notes). For $n \geq 0$:

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n$$

Let's prove this by counting sets, rather than algebraic manipulation.

The RHS, 2^n, is the number of subsets of an n-element set.

On the LHS, for $0 \leq k \leq n$, $\binom{n}{k}$ is the number of k-element subsets of S. And addition corresponds to "or", so the LHS is the number of ways of choosing a single subset of S: choosing one of the
the $\binom{n}{0}$ 0-element subsets or
the $\binom{n}{1}$ 1-element subsets or
$\vdots$
the $\binom{n}{n}$ n-element subsets.

Lecture: May 12

Example 2 (Example 1.7 in the notes). For $k, n \geq 1$:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

The LHS, $\binom{n}{k}$, is the number of k-element subsets of an n-element set S. WLOG suppose $S = \{1, 2, \ldots, n\}$.

Notice that $\binom{n-1}{k}$ is the number of k-element subsets of $S' = \{1, 2, \ldots, n-1\}$ and $\binom{n-1}{k-1}$ is the number of $(k-1)$-element subsets of S'. Recall our intuition that addition corresponds to "or".

Any k-element subset X of S either includes n or does not include n. If $n \notin X$, then X is a subset of size k of a set of size $n-1$. If $n \in X$, then $X \setminus \{n\}$ is a subset of size $k-1$ of a set of size $n-1$. Thus LHS=RHS.

This formula allows us to recursively compute $\binom{n}{k}$, leading to the famous Pascal's triangle:

$$\begin{array}{ccccccccccc}
 & & & & & \binom{0}{0} & & & & & \\
 & & & & \binom{1}{0} & & \binom{1}{1} & & & & \\
 & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & \\
 & & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & \\
 & \binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & \\
\binom{5}{0} & & \binom{5}{1} & & \binom{5}{2} & & \binom{5}{3} & & \binom{5}{4} & & \binom{5}{5}
\end{array}
=
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}$$

1.3 Multisets (§1.1.4)

Suppose I have a bag containing 8 marbles, each of which is either red, green, or blue. How many different bag contents are possible?

First question: how can I represent the contents of a bag? We could have for example $n_1 = 2$ red, $n_2 = 3$ green, and $n_3 = 3$ blue marbles

- We might be tempted to write this as the set $\{R, R, G, G, G, B, B, B\}$, but this isn't proper notation: sets don't have duplicates, so this formally collapses to $\{R, G, B\}$, losing the multiplicity information.
- Make up some notation that preserves multiplicity, like: $[\![R, R, G, G, G, B, B, B]\!]$ or $\langle\langle R^{\times 2}, G^{\times 3}, B^{\times 3}\rangle\rangle$
- But really all I need to do is keep track of the multiplicities. Fix an order – say R then G then B – and use a tuple to record the multiplicities: $(2, 3, 3)$.

In other words to count the number of bags containing 8 marbles each of which is either red, green, or blue, we need to count the number of 3-tuples of integers (n_1, n_2, n_3) such that $n_1 + n_2 + n_3 = 8$ and $n_1, n_2, n_3 \geq 0$.

Definition 1 (Definition 1.8 in the notes)**.** Let $n \geq 0$, $t \geq 1$ be integers. A **multiset of size** n **with elements of** t **types** is a sequence of non-negative integers $(m_1, \ldots, m_t)$ such that $m_1 + \cdots + m_t = n$.

Theorem 4 (Theorem 1.9 in the notes)**.** *For any $n \geq 0$, $t \geq 1$, the number of n-element multisets with elements of t types is*

$$\binom{n+t-1}{t-1}$$

Proof. We know that $\binom{n+t-1}{t-1}$ is the number of $(t-1)$-element subsets of an $(n+t-1)$-element set. We'll show how to translate between $(t-1)$-element subsets of an $(n+t-1)$-element set and n-element multisets with t types.

Let's write down a row of $n + t - 1$ circles:

$$\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ$$

Now, let's choose a $(t-1)$-element subset of these and cross them out:

$$\circ\circ\times\circ\circ\circ\times\circ\circ\circ$$

Our row now has n circles, grouped into t segments each containing 0 or more consecutive circles. Let m_i be the length of the ith segment of consecutive circles. Then $m_1 + \cdots + m_t = n$, so $(m_1, \ldots, m_t)$ is an n-element multiset with t types.

Now let's go the other way. Let $(m_1, \ldots, m_n)$ be an n-element multiset with t types. Write down a sequence of m_1 circles, then an X, then m_2 circles, then an X, and so on, finishing with an X and m_t

circles. We'll have written down $n+t-1$ symbols, with the $t-1$ X's indicating a $(t-1)$ element subset. □

This is an example of a bijection.

1.4 Bijections (§1.1.5)

Definition 2 (Definition 1.10 in the notes)**.** Let A and B be sets, and let $f : A \to B$.

- f is **surjective (onto)** if, for every $b \in B$, there exists an $a \in A$ such that $f(a) = b$.
- f is **injective (one-to-one)** if, for every $a, a' \in A$, if $f(a) = f(a')$ then $a = a'$
- f is **bijective** if it is both surjective and injective.

Proposition 1 (Proposition 1.11 in the notes)**.** *A function $f : A \to B$ is a bijection if and only if f has an inverse, namely if there exists a function $g : B \to A$ such that $g(f(a)) = a$ for all $a \in A$ and $f(g(b)) = b$ for all $b \in B$.*

If there exists a bijection between two sets A and B, we write $A \rightleftharpoons B$.

Corollary 1. *If $A \rightleftharpoons B$ and at least one of A or B is finite, then both are finite and $|A| = |B|$.*

Lecture: May 15

We've implicitly used bijections already to prove some of our identities.

Example 3 (Example 1.7 in the notes). For $k, n \geq 1$:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Let $S = \{1, \dots n\}$ and $S' = \{1, \dots, n-1\}$.

Let $X = \{k\text{-element subsets of } S\}$ and $Y = \{(k-1)\text{-element or } k\text{-element subsets of } S'\}$. By definition have $|A| = \binom{n}{k}$ and $|B| = \binom{n-1}{k} + \binom{n-1}{k-1}$.

We now prove $A \rightleftharpoons B$. Define $f : A \to B$ by

$$f(u) = \begin{cases} u, & \text{if } n \notin u, \\ x \setminus \{n\}, & \text{if } n \in u. \end{cases}$$

Define $g : B \to A$ by

$$g(v) = \begin{cases} v, & \text{if } |v| = k, \\ v \cup \{n\}, & \text{if } |v| = k-1. \end{cases}$$

For all $u \in A$, $g(f(u)) = u$. For all $v \in B$, $f(g(v)) = v$. Thus f is a bijection, so $A \rightleftharpoons B$ and $|A| = |B|$.

Exercise 1. Show that $\binom{n}{k} = \binom{n}{n-k}$ for integers $n \geq k \geq 0$.

Obviously we can solve this algebraically using Theorem 1.5 Can you find a combinatorial proof? Hint: find a bijection between subsets of size k and subsets of size $n-k$.

2 Generating Series (§2)

Definition 3. An **formal power series** is an expression of the form

$$G(x) = \sum_{n=0}^{\infty} g_n x^n$$

in which the coefficients $(g_0, g_1, g_2, \dots)$ are a sequence of integers.

We use x as an **indeterminate** for which we do not normally substitute in any particular value. Thus, unlike in calculus, for example, we are not concerned whether a formal power series converges; all we require is that each coefficient is finite.

Example:

$$G(x) = 0 - 1x + 2x^2 - 3x^3 + 4x^4 - \cdots = \sum_{n=0}^{\infty} (-1)^n n x^n$$

We can add and multiply formal power series like polynomials: adding them (term by term) or multiplying them (collecting like powers).

$$\begin{aligned}
F(x) &= \sum_{n\geq 0} f_n x^n = f_0 + f_1 x + f_2 x^2 + \dots \\
G(x) &= \sum_{n\geq 0} g_n x^n = g_0 + g_1 x + g_2 x^2 + \dots \\
(F+G)(x) &= \sum_{n\geq 0} (f_n + g_n) x^n = (f_0 + g_0) + (f_1 + g_1)x + (f_2 + g_2)x^2 + \dots \\
(F \cdot G)(x) &= \sum_{n\geq 0} \left(\sum_{k=0}^{n} f_k g_{n-k} \right) x^n \\
&= f_0 g_0 + (f_0 g_1 + f_1 g_0)x + \dots
\end{aligned}$$

Example:

$$\begin{aligned}
&(1 + 3x^2 + 5x^4 + \dots) + x(2 + 4x^2 + 6x^4 + \dots) \\
&= (1 + 3x^2 + 5x^4 + \dots) + (2x + 4x^3 + 6x^5 + \dots) \\
&= 1 + 2x + 3x^2 + 4x^3 + 5x^4 + 6x^5 + \dots)
\end{aligned}$$

or equivalently

$$\begin{aligned}
&\left(\sum_{n\geq 0}(2n+1)x^{2n}\right) + x\left(\sum_{n\geq 0}(2n+2)x^{2n}\right)\\
&= \left(\sum_{n\geq 0}(2n+1)x^{2n}\right) + \left(\sum_{n\geq 0}(2n+2)x^{2n+1}\right)\\
&= \sum_{n\geq 0}(2n+1)x^{2n} + (2n+2)x^{2n+1}\\
&= \sum_{n\geq 0}(n+1)x^{n}
\end{aligned}$$

Example 4 (Example 2.1 in the notes). Let $G(x) = 1 + x + x^2 + x^3 + \cdots = \sum_{n\geq 0} x^n$, the **geometric series**. Then $xG(x) = x + x^2 + x^3 + x^4 + \cdots = x\sum_{n\geq 0} x^n$. So

$$G(x) - xG(x) = 1 + x + x^2 + x^3 + \cdots - x - x^2 - x^3 - x^4 - \cdots = 1$$

In other words,

$$(1-x)G(x) = 1$$

so

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

We can sometimes invert them: if $F(x)$ and $G(x)$ are formal power series such that $F(x)G(x) = 1$, then we write $G(x) = F(x)^{-1}$ or $G(x) = \frac{1}{F(x)}$.

Proposition 2. *The inverse of $F(x) = \sum_{n\geq 0} f_n x^n$ exists if and only if $f_0 \neq 0$.*

Lecture: May 17

2.1 Binomial theorem (§2.1)

Theorem 5 (The Binomial Theorem, Theorem 2.2 in the notes)**.** *For any $n \in \mathbb{N}$,*

$$(1+x)^n = \sum_{k=0}^{n}\binom{n}{k}x^k = \sum_{k\geq 0}\binom{n}{k}x^k$$

You've probably seen before an algebraic proof using induction on n. Let's try for a combinatorial proof.

Proof. Let $\mathcal{P}(n)$ be the power set of $\{1,\ldots,n\}$, i.e., the set of all subsets of $\{1,\ldots,n\}$. We saw before that there's a correspondence between choosing a subset S of an n-element set and choosing an "indicator vector" indicating whether an element v is in S or out. E.g.: For $\{1,2,3\}$ the subset $S = \{1,2\}$ corresponds to the indicator vector (in, in, out). But let's using 0 and 1 to represent out and in. So we have a bijection between $\mathcal{P}(n)$ and $\{0,1\}^n$, where S is a subset and $\vec{a} = (a_1, a_2, \ldots, a_n)$

is the indicator vector with $a_i = 1$ if $i \in S$ and $a_i = 0$ if $i \notin S$. Most importantly, note that the size of the subset is $|S| = a_1 + a_2 + \cdots + a_n$.

Now let's introduce an indeterminate x. For every subset S, if $\vec{a}$ is the corresponding indicator vector, then

$$x^{|S|} = x^{a_1+a_2+\cdots+a_n}$$

Moreover because of the bijection, summing over all subsets is equivalent to summing overall indicator vectors:

$$\sum_{S \in \mathcal{P}(n)} x^{|S|} = \sum_{\vec{a} \in \{0,1\}^n} x^{a_1+a_2+\cdots+a_n}$$

Now let's simplify each side separately.

On the LHS, apply Theorem 1.5: there are $\binom{n}{k}$ k-element subsets of an n-element set for each $0 \le k \le n$:

$$LHS = \sum_{S \in \mathcal{P}(n)} x^{|S|} = \sum_{k=0}^{n} \sum_{S \in \mathcal{P}(n), |S|=k} x^{|S|} = \sum_{k=0}^{n} \binom{n}{k} x^k$$

On the RHS, summing over all $\vec{a} \in \{0,1\}^n$ can be broken up into summing over all $a_1 \in \{0,1\}$, and all $a_2 \in \{0,1\}$, and so on:

$$\begin{aligned} RHS &= \sum_{\vec{a} \in \{0,1\}^n} x^{a_1+a_2+\cdots+a_n} \\ &= \sum_{a_1=0}^{1} x^{a_1} \sum_{a_2=0}^{1} x^{a_2} \cdots \sum_{a_n=0}^{1} x^{a_n} \\ &= (1+x)^n \end{aligned}$$

□

Theorem 6 (The negative binomial theorem, a.k.a. the binomial series theorem, Theorem 2.4 in the notes). *If $t \ge 1$, then*

$$(1-x)^{-t} = \sum_{n \ge 0} \binom{n+t-1}{t-1} x^n$$

Proof. Another combinatorial proof.

We recognize $\binom{n+t-1}{t-1}$ as the number of n-element multisets with t types (denote $|\mathcal{M}(n,t)|$. That is, non-negative integer sequences $(m_1, m_2, \ldots, m_t)$ such that $m_1 + m_2 + \cdots + m_t = n$. Let's consider $\mathcal{M}(t)$ to be the set of all multisets (with any number of elements) of t types. We can see that there is a bijection between $\mathcal{M}(t)$ and $\mathbb{N}^t$: every multiset $\mu \in \mathcal{M}(t)$ corresponds to a non-negative integer sequence $(m_1, m_2, \ldots, m_t)$, and vice versa. Moreover, the number of elements in the multiset μ is $|\mu| = m_1 + m_2 + \cdots + m_t$

So:

$$\begin{aligned}
\sum_{n\geq 0}\binom{n+t-1}{t-1}x^n &= \sum_{n\geq 0}|\mathcal{M}(n,t)|x^n && \text{(by Theorem 1.9)}\\
&= \sum_{\mu\in\mathcal{M}(t)} x^{|\mu|} && \text{(another way of counting all multisets)}\\
&= \sum_{(m_1,m_2,\ldots,m_t)\in\mathbb{N}^t} x^{m_1+m_2+\cdots+m_t} && \text{(bijection)}\\
&= \sum_{m_1\geq 0}\sum_{m_2\geq 0}\cdots\sum_{m_t\geq 0} x^{m_1+m_2+\cdots+m_t} && \text{(Cartesian product)}\\
&= \sum_{m_1\geq 0} x^{m_1}\sum_{m_2\geq 0} x^{m_2}\cdots\sum_{m_t\geq 0} x^{m_t} && \text{(multiplication)}\\
&= \left(\frac{1}{1-x}\right)^t && \text{(geometric series)}
\end{aligned}$$

□

Lecture: May 19

2.2 Generating series (§2.1.1)

We should practice manipulating formal power series.

Example 5. Write $\left(\frac{x^2}{1+3x}\right)^4$ as a formal power series.

$$\begin{aligned}
\left(\frac{x^2}{1+3x}\right)^4 &= x^8(1-(-3x))^{-4}\\
&= x^8\sum_{n\geq 0}\binom{n+4-1}{4-1}(-3x)^n && \text{(by negative binomial theorem)}\\
&= \sum_{n\geq 0}\binom{n+3}{3}(-3)^n x^{n+8}\\
&= \sum_{n\geq 8}\binom{n-5}{3}(-3)^{n-8}x^n
\end{aligned}$$

Definition 4 (Coefficient extraction, Definition 2.8 in the notes)**.** Let $G(x) = \sum_{n\geq 0} g_n x^n$ be a formal power series. For $k\in\mathbb{N}$, define

$$[x^k]G(x) = g_k$$

i.e., $[x^k]$ extracts the coefficient of x^k in $G(x)$.

Example 6. Continuing our example above, we have that

$$[x^k]\left(\frac{x^2}{1+3x}\right)^4 = [x^k]\sum_{n\geq 8}\binom{n-5}{3}(-3)^{n-8}x^n = \begin{cases} 0, & \text{if } 0\leq k<8,\\ \binom{n-5}{3}(-3)^{n-8}, & \text{if } k\geq 8\end{cases}$$

Some simple rules about coefficient extraction:

- $[x^k](aF(x) + bG(x)) = a[x^k]F(x) + b[x^k]G(x)$
- $[x^k](x^\ell F(x)) = [x^{k-\ell}]F(x)$
- $[x^k](F(x)G(x)) = \sum_{\ell=0}^{k} \left([x^\ell]F(x)\right)\left([x^{k-\ell}]G(x)\right)$

We will use formal power series to encode counting information about a set, in which case we call it a generating series.

Example 7. Let $\mathcal{M} = \{\text{January, February, March, \ldots, December}\}$. Let

$$\mathcal{M}_n = \{\alpha \in \mathcal{M} : \alpha \text{ has exactly } n \text{ days (no leap years)}\}$$

For example:

$$\mathcal{M}_0 = \emptyset \quad , \quad \mathcal{M}_{28} = \{\text{February}\} \quad , \quad \mathcal{M}_{30} = \{\text{April, June, September, November}\}$$

So

$$\sum_{n\geq 0} |\mathcal{M}_n| x^n = x^{28} + 4x^{30} + 7x^{31}$$

I could also come up with the same summation in a different way:

$$\sum_{\alpha\in\mathcal{M}} x^{(\text{\# days in } \alpha)} = x^{31} + x^{28} + x^{31} + x^{30} + \cdots + x^{31} = x^{28} + 4x^{30} + 7x^{31}$$

Definition 5 (Definition 2.5 in the notes)**.** Let S be a set. A **weight function** is a function $\omega : S \to \mathbb{N}$ if, for every $n \in \mathbb{N}$, the number of elements of S of weight n is finite, i.e., $\{\alpha \in S : \omega(\alpha) = n\}$ is finite.

Definition 6 (Definition 2.6 in the notes)**.** Let S be a set and ω be a weight function on S. The **generating series of S with respect to** ω is

$$\Phi_S^\omega(x) = \Phi_S(x) = \sum_{\alpha\in S} x^{\omega(\alpha)}$$

For our example above, $\omega(\alpha)$ counts the number of days in α, and we have that the generating series of $\mathcal{M}$ with respect to α is

$$\Phi_{\mathcal{M}}(x) = \sum_{\alpha\in\mathcal{M}} x^{\omega(\alpha)} = x^{28} + 4x^{30} + 7x^{31}$$

Proposition 3 (Proposition 2.7 in the notes)**.** *Let ω be a weight function on a set S. Then*

$$\Phi_S(x) = \sum_{n\geq 0} \underbrace{|\{\alpha \in S : \omega(\alpha) = n\}|}_{\text{\# elements of weight } n \text{ in } S} x^n$$

Equivalently, for each $k \geq 0$,

$$[x^k]\Phi_S(x) = |\{\alpha \in S : \omega(\alpha) = k\}|$$

(This proposition is a generalization of the two equivalent sums I wrote for the months: either iterating over all integers n and counting the number of months with that many days ($\sum_{n\geq 0}|\mathcal{M}_n|x^n$) or iterating over all months and recording one term for each month ($\sum_{\alpha\in\mathcal{M}} x^{(\text{\# of days in } \alpha)}$), then collecting like terms.)

Proof. Let $S_n = \{\alpha \in S : \omega(\alpha) = n\}$. Then $S = \cup_{n\geq 0} S_n$.

$$\Phi_S(x) = \sum_{\alpha\in S} x^{\omega(\alpha)} = \sum_{n\geq 0}\left(\sum_{\alpha\in S_n} x^{\omega(\alpha)}\right) = \sum_{n\geq 0}\left(\sum_{\alpha\in S_n} x^n\right) = \sum_{n\geq 0}|S_n|x^n$$

□

Example 8. Write the generating series for the set S of all binary strings with respect to weight function ω which records the length of the string.

$$\begin{aligned}\Phi_S(x) &= \sum_{\alpha\in S} x^{(\text{length of } \alpha)} \\ &= \sum_{n\geq 0}(\text{\# of binary strings of length } n)x^n && \text{(Proposition 2.7)} \\ &= \sum_{n\geq 0} 2^n x^n \\ &= \frac{1}{1-2x} && \text{(generalization of geometric series)}\end{aligned}$$

Example 9. Write the generating series for the set

$$S = \{\text{subsets of } \{1,2,\ldots,t\}\} \quad , \quad \omega(\alpha) = |\alpha|$$

$$\begin{aligned}\Phi_S(x) &= \sum_{\alpha\subseteq\{1,\ldots,t\}} x^{|\alpha|} \\ &= \sum_{n\geq 0}(\text{\# of } n\text{-element subsets of } \{1,\ldots,t\})x^n && \text{(Proposition 2.7)} \\ &= \sum_{n\geq 0}\binom{t}{n}x^n \\ &= (1+x)^t && \text{(Binomial theorem)}\end{aligned}$$

Lecture: May 23

2.3 Sum and product lemmas (§2.2.2)

Example 10. Consider breakfast foods and the weight function that corresponds to their cost. Let

$$\begin{array}{rccccccc} S_1 = \{ & \text{omelette,} & \text{waffles,} & \text{pancakes,} & \text{eggs,} & \text{cereal} & \} \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ \omega_1 : S_1 \to \mathbb{N} & 10 & 10 & 8 & 8 & 5 & \end{array}$$

The generating series for breakfast mains with respect to cost:

$$\Phi_{S_1}^{\omega_1}(x) = \sum_{\alpha \in S_1} x^{\omega_1(\alpha)} = x^5 + 2x^8 + 2x^{10}$$

$$\Phi_{S_1}^{\omega_1}(x) = \sum_{\alpha \in S_1} x^{\omega_1(\alpha)} = \sum_{n \geq 0} |\{\alpha \in S_1 : \omega_1(\alpha) = n\}| x^n = x^5 + 2x^8 + \underbrace{2}_{\text{Number of things of this cost}} x^{\underbrace{10}_{\text{the cost}}}$$

Let

$$\begin{array}{rcccl} S_2 = \{ & \text{bacon,} & \text{hashbrowns,} & \text{toast} & \} \\ & \downarrow & \downarrow & \downarrow & \\ \omega_2 : S_2 \to \mathbb{N} & 5 & 4 & 3 & \end{array}$$

The generating series for breakfast sides with respect to cost:

$$\Phi_{S_2}^{\omega_2}(x) = \sum_{\alpha \in S_2} x^{\omega_2(\alpha)} = x^3 + x^4 + x^5$$

$$\begin{array}{rccccccccl} S_1 \cup S_2 = \{ & \text{omelette,} & \text{waffles,} & \text{pancakes,} & \text{eggs,} & \text{cereal,} & \text{bacon,} & \text{hashbrowns,} & \text{toast} & \} \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ & 10 & 10 & 8 & 8 & 5 & 5 & 4 & 3 & \end{array}$$

The generating series for any breakfast item (main or side) with respect to cost:

$$\Phi_{S_1}^{\omega_1}(x) + \Phi_{S_2}^{\omega_2}(x) = \sum_{\alpha \in S_1 \cup S_2} x^{\omega(\alpha)} = x^3 + x^4 + 2x^5 + 2x^8 + 2x^{10} = \Phi_{S_1 \cup S_2}^{\omega}(x)$$

$\Phi_{S_1 \cup S_2}(x)$ counts the number of breakfast items of each price.

$$\begin{aligned} \Phi_{S_1}^{\omega_1}(x) \cdot \Phi_{S_2}^{\omega_1}(x) &= (x^5 + 2x^8 + 2x^{10})(x^3 + x^4 + x^5) \\ &= x^8 + x^9 + x^{10} + 2x^{11} + 2x^{12} + 2x^{13} + 2x^{13} + 2x^{14} + 2x^{15} \\ &= x^8 + x^9 + x^{10} + 2x^{11} + 2x^{12} + 4x^{13} + 2x^{14} + 2x^{15} \end{aligned}$$

For example, $2x^{12}$ counts the number of ways of choosing a main and a side with total price 12. This is the generating series for breakfast combos with respect to price. This can be denoted

$$\Phi_{S_1 \times S_2}^{\omega}(x)$$

with joint weight function

$$\omega(\alpha_1, \alpha_2) = \omega_1(\alpha_1) + \omega_2(\alpha_2)$$

Lemma 1 (Sum Lemma, Lemma 2.10 in the notes). *Let S_1, S_2 be disjoint sets and let ω be a weight function on $S_1 \cup S_2$. Then*

$$\Phi_{S_1}(x) + \Phi_{S_2}(x) = \Phi_{S_1 \cup S_2}(x)$$

Proof. Main idea: sums correspond to disjoint union.

$$\begin{aligned}
\Phi_{S_1}(x) + \Phi_{S_2}(x) &= \sum_{\alpha \in S_1} x^{\omega(\alpha)} + \sum_{\alpha \in S_2} x^{\omega(\alpha)} && \text{(by definition)} \\
&= \sum_{\alpha \in S_1 \cup S_2} x^{\omega(\alpha)} && \text{(since sets disjoint)} \\
&= \Phi_{S_1 \cup S_2}(x) && \text{(by definition)}
\end{aligned}$$

□

We can generalize to the infinite case:

Lemma 2 (Infinite sum Lemma, Lemma 2.11 in the notes). *Let $S_0, S_1, S_2, \ldots$ be disjoint sets with union S and let ω be a weight function on S. Then*

$$\Phi_S(x) = \sum_{n \geq 0} \Phi_{S_n}(x)$$

(Probably omit this proof during the lecture.)

Proof. Main idea: infinite sums correspond to infinite disjoint union.

$$\begin{aligned}
\sum_{n \geq 0} \Phi_{S_n}(x) &= \sum_{n \geq 0} \sum_{\alpha \in S_n} x^{\omega(\alpha)} \\
&= \sum_{\alpha \in S_0 \cup S_1 \cup \ldots} x^{\omega(\alpha)} \\
&= \sum_{\alpha \in S} x^{\omega(\alpha)} \\
&= \Phi_S(x)
\end{aligned}$$

□

Lemma 3 (Product Lemma, Lemma 2.12 in the notes). *Let S_1, S_2 be sets and let ω_1 and ω_2 be weight functions on S_1 and S_2 respectively. Then*

$$\Phi_{S_1}^{\omega_1}(x) \Phi_{S_2}^{\omega_2}(x) = \Phi_{S_1 \times S_2}^{\omega}(x)$$

where ω is the weight function on $S_1 \times S_2$ defined by $\omega(\alpha_1, \alpha_2) = \omega_1(\alpha_1) + \omega_2(\alpha_2)$.

Proof. Main idea: products correspond to Cartesian products.

$$\begin{aligned}
\Phi^{\omega_1}_{S_1}(x)\Phi^{\omega_2}_{S_2}(x) &= \sum_{\alpha_1\in S_1} x^{\omega_1(\alpha_1)} \sum_{\alpha_2\in S_2} x^{\omega_2(\alpha_2)} && \text{(by definition)}\\
&= \sum_{\alpha_1\in S_1}\sum_{\alpha_2\in S_2} x^{\omega_1(\alpha_1)+\omega_2(\alpha_2)}\\
&= \sum_{(\alpha_1,\alpha_2)\in S_1\times S_2} x^{\omega(\alpha_1,\alpha_2)}\\
&= \Phi^{\omega}_{S_1\times S_2}(x)
\end{aligned}$$

□

Example 11. Let $S = \{2, 4, 6, 8, \dots\}$. How many pairs (a, b) are there with $a, b \in S$ and $a + b = 50$?

Let $\omega(\alpha) = \alpha$ for each $\alpha \in S$. Then the number of pairs $(a, b) \in S \times S$ with $\omega(a) + \omega(b) = 50$ is

$$\begin{aligned}
[x^{50}]\Phi_{S\times S}(x) &= [x^{50}]\left(\Phi_S(x)\times\Phi_S(x)\right) && \text{(by Product Lemma)}\\
&= [x^{50}](x^2+x^4+x^6+\dots)^2 && \text{(substitute generating series for } \Phi_S(x))\\
&= [x^{50}](x^2(1+x^2+x^4+\dots))^2 && \text{(common factor)}\\
&= [x^{50}]\frac{x^4}{(1-x^2)^2} && \text{(geometric series)}\\
&= [x^{46}](1-x^2)^{-2} && \text{(coefficient extraction rules)}\\
&= [x^{46}]\sum_{n\geq 0}\binom{n+2-1}{2-1}(x^2)^n && \text{(negative binomial theorem)}\\
&= \binom{23+2-1}{2-1} && \text{(taking } n=23 \text{ to get } x^{46})\\
&= 24
\end{aligned}$$

Lecture: May 24

Definition 7 (Star operator)**.** Let A be a set. Define

$$A^* = \cup_{k\geq 0} A^k = \{\text{ all tuples of elements of } A\}$$

Example 12.

$$\{0,1\}^* = \{\underbrace{\overbrace{()}^{\varepsilon}}_{A^0}, \underbrace{\overbrace{(0)}^{0}, \overbrace{(1)}^{1}}_{A^1}, \underbrace{\overbrace{(0,0)}^{00}, \overbrace{(0,1)}^{01}, \overbrace{(1,0)}^{10}, \overbrace{(1,1)}^{11}}_{A^2}, \underbrace{\overbrace{(0,0,0)}^{000}, \dots}_{A^3,\dots}\}$$

Example 13.

$$\{1,2,3,\dots\}^* = \{\underbrace{()}_{A^0}, \underbrace{(1),(2),(3),\dots}_{A^1}, \underbrace{(1,1),(1,2),\dots}_{A^2}, \underbrace{(1,1,1),\dots}_{A^3,\dots}\}$$

Given a weight function ω on A, we can define ω^* on A^* such that $\omega^*((a_1, a_2, \dots, a_k)) = \sum_{i=1}^k \omega(a_i)$. This is well defined provided no elements of A have weight 0.

Example 14. Let $S = \{0, 1\}$ and ω count length, i.e., $\omega(0) = \omega(1) = 1$. Then

$$\begin{aligned}
\omega^*((1,1,0,1)) &= \omega(1) + \omega(1) + \omega(0) + \omega(1) && = 4 \\
\omega^*((1)) &= \omega(1) && = 1 \\
\omega^*(())) &= \text{(empty sum)} && = 0
\end{aligned}$$

Then ω^* also counts length.

How are $\Phi_S(x)$ and $\Phi_{S^*}(x)$ related?

$$\begin{aligned}
\Phi_S(x) &= \sum_{a \in S} x^{\omega(a)} = x^{\omega(0)} + x^{\omega(1)} = 2x^1 \\
\Phi_{S^*}(x) &= \Phi_{\{(),(0),(1),(0,0),(0,1),(1,0),(1,1),(0,0,0),\dots\}}(x) \\
&= 1x^0 + 2x^1 + 4x^2 + 8x^3 + \dots \\
&= \frac{1}{1-2x} && \text{(geometric series)} \\
&= \frac{1}{1-\Phi_S(x)}
\end{aligned}$$

Is this a coincidence?

Lemma 4 (String Lemma, 2.14 in the notes)**.** *Let A be a set with weight function ω such that no elements of A have weight 0. Then*

$$\Phi_{A^*}(x) = \frac{1}{1-\Phi_A(x)}$$

Proof.

$$\begin{aligned}
\Phi_{A^*}(x) &= \Phi_{A^0 \cup A_1 \cup A_2 \cup \dots}(x) && \text{(definition of } A^*\text{)} \\
&= \sum_{n \geq 0} \Phi_{A^n}(x) && \text{(infinite sum lemma)} \\
&= \sum_{n \geq 0} (\Phi_A(x))^n && \text{(product lemma)} \\
&= \frac{1}{1-\Phi_A(x)} && \text{(geometric series)}
\end{aligned}$$

□

2.4 Compositions (§2.3)

Definition 8. A **composition** is a finite sequence of *positive* integers

$$\gamma = (c_1, c_2, \dots, c_k)$$

Each $c_i \in \mathbb{Z}_{>0}$ is called a **part**. The **length** of the composition is the number of parts, $\ell(\gamma) = k$. The **size** of the composition is the sum of the parts, $|\gamma| = c_1 + c_2 + \cdots + c_k$. If s is the size of γ, then we say that γ is a composition of s.

Example 15.
- $()$ is a composition of 0 with 0 parts
- $(1, 2, 3, 4)$ is a composition of 10 with 4 parts
- $(1, 1, 2)$ is a composition of 4 with 3 parts

Theorem 7 (Theorem 2.17b in the notes)**.** *The generating series for all integer compositions with respect to size is*

$$\Phi_{\textit{compositions}}(x) = \frac{1-x}{1-2x} = 1 + \frac{x}{1-2x}$$

Example 16. How many compositions are there of 123?

$$\begin{aligned}
\text{\# of compositions of 123} &= [x^{123}]\Phi_{\text{compositions}}(x) \\
&= [x^{123}]\frac{1-x}{1-2x} \\
&= [x^{123}](1-x)\sum_{n\geq 0}(2x)^n \\
&= [x^{123}]\sum_{n\geq 0}(2x)^n - [x^{122}]\sum_{n\geq 0}(2x)^n \\
&= 2^{123} - 2^{122} = 2^{122}
\end{aligned}$$

Proof of Theorem 2.17b.

$$\begin{aligned}
\Phi_{\text{compositions}}(x) &= \Phi_{(\mathbb{Z}_{>0})^*}(x) \\
&= \frac{1}{1-\Phi_{\mathbb{Z}_{>0}}(x)} && \text{(string lemma)} \\
&= \frac{1}{1-\sum_{n\in\mathbb{Z}_{>0}} x^n} \\
&= \frac{1}{1-x\sum_{n\geq 0} x^n} \\
&= \frac{1}{1-x(1-x)^{-1}} && \text{(geometric series)} \\
&= \frac{1-x}{1-x-x} = \frac{1-x}{1-2x}
\end{aligned}$$

□

Lecture: May 26

Theorem 8 (Theorem 2.17c in the notes)**.** *For $n \in \mathbb{N}$, the number of compositions of size n is*

$$\begin{cases} 1, & \textit{if } n = 0, \\ 2^{n-1}, & \textit{if } n \geq 1 \end{cases}$$

Example 17. How many compositions are there of 123 where each part is either 1 or 3?

Let $R = \{1, 3\}$. Note $\Phi_R(x) = x^1 + x^3$.

The set of compositions with each part being either 1 or 3 is R^*.

We want

$$\begin{aligned}
[x^{123}]\Phi_{R^*}(x) &= [x^{123}]\frac{1}{1-\Phi_R(x)} && \text{(string lemma)} \\
&= [x^{123}]\frac{1}{1-x-x^3}
\end{aligned}$$

For now, we'll leave the solution in this form, and see how to write a recurrence relation for this later.

Example 18. Find the generating series for compositions where each part is odd?

Let $S = \{1, 3, 5, 7, \dots\}$. Note

$$\Phi_S(x) = x^1 + x^3 + x^5 + \cdots = x(1 + x^2 + x^4 + \dots) = \frac{x}{1-x^2}$$

The set of compositions with only odd parts is S^*.

We want

$$\begin{aligned}
\Phi_{S^*}(x) &= \frac{1}{1-\Phi_S(x)} && \text{(string lemma)} \\
&= \frac{1}{1-\frac{x}{1-x^2}} \\
&= \frac{1-x^2}{1-x-x^2} \\
&= 1 + \frac{x}{1-x-x^2}
\end{aligned}$$

Example 19. Find the generating series for compositions where each part is 2 or greater?

Let $T = \{2, 3, 4, 5, \dots\}$. Note

$$\Phi_T(x) = x^2 + x^3 + x^4 + \cdots = x^2(1 + x + x^2 + \dots) = \frac{x^2}{1-x}$$

The set of compositions where each part is 2 or greater is T^*.

We want

$$\begin{aligned}\Phi_{T^*}(x) &= \frac{1}{1-\Phi_T(x)} \qquad \text{(string lemma)}\\ &= \frac{1}{1-\frac{x^2}{1-x}}\\ &= \frac{1-x}{1-x-x^2}\\ &= 1+\frac{x^2}{1-x-x^2}\end{aligned}$$

The generating series in these last two examples ($\Phi_{S^*}(x)$ and $\Phi_{T^*}(x)$ look rather similar. In fact, there's a stronger relation.

Claim 1. The number of compositions of n where each part is 2 or greater is equal to the number of compositions of $n-1$ where each part is odd.

Proof. We want to show

$$[x^{n-1}]\Phi_{S^*}(x) = [x^n]Phi_{T^*}(x)$$

$$\begin{aligned}LHS &= [x^{n-1}]\Phi_{S^*}(x)\\ &= [x^{n-1}]\left(1+\frac{x}{1-x-x^2}\right)\\ &= [x^{n-1}]1+[x^{n-1}]\frac{x}{1-x-x^2}\\ &= 0+[x^{n-2}]\frac{1}{1-x-x^2}\\ RHS &= [x^n]\Phi_{T^*}(x)\\ &= [x^n]\left(1+\frac{x^2}{1-x-x^2}\right)\\ &= [x^n]1+[x^n]\frac{x^2}{1-x-x^2}\\ &= 0+[x^{n-2}]\frac{1}{1-x-x^2}\\ &= LHS\end{aligned}$$

□

Since these are finite sets that are the same size, there exists a bijection between them. Perhaps there's a deeper connection... can we find a "natural" bijection that explains this relationship?

Compositions of $n = 7$ with parts ≥ 2	Compositions of $n - 1 = 6$ with odd parts
(7)	$(1, 5)$
$(5, 2)$	$(5, 1)$
$(2, 5)$	$(1, 1, 1, 3)$
$(4, 3)$	$(1, 1, 3, 1)$
$(3, 4)$	$(1, 3, 1, 1)$
$(3, 2, 2)$	$(3, 1, 1, 1)$
$(2, 3, 2)$	$(3, 3)$
$(2, 2, 3)$	$(1, 1, 1, 1, 1, 1)$

Can we see a systematic way to convert from one to the other? This is a bit tricky to come up with, but here's one way:

1. Take a composition of 7 with parts ≥ 2.
2. Subtract 1 from the last part. (Now it's a composition of 6 with all but the last part ≥ 2.)
3. For each even part $2k$, split it into $(1, 2k - 1)$. (Now it's a composition of 6 with all odd parts.)

Compositions of $n = 7$ with parts ≥ 2		Compositions of $n - 1 = 6$ with odd parts
(7)	$\to (6)$	$\to (1, 5)$
$(5, 2)$	$\to (5, 1)$	$\to (5, 1)$
$(2, 5)$	$\to (2, 4)$	$\to (1, 1, 1, 3)$
$(4, 3)$	$\to (4, 2)$	$\to (1, 3, 1, 1)$
$(3, 4)$	$\to (3, 3)$	$\to (3, 3)$
$(3, 2, 2)$	$\to (3, 2, 1)$	$\to (3, 1, 1, 1)$
$(2, 3, 2)$	$\to (2, 3, 1)$	$\to (1, 1, 3, 1)$
$(2, 2, 3)$	$\to (2, 2, 2)$	$\to (1, 1, 1, 1, 1, 1)$

To prove that this is a bijection, we'd have to either (a) prove that it's injective and surjective, or (b) write down a function g and prove that g and f are inverses of each other.

Lecture: May 29

3 Binary Strings (§3)

Definition 9. A **binary string** of length $n \geq 0$ is a finite sequence $\sigma = b_1 b_2 \dots b_n$ where each **bit** $b_i \in \{0, 1\}$.

Examples: ϵ (the empty string, which has length 0), 0, 1, 00, 01, 10, 11, 000, etc.

Binary strings correspond to elements of the set

$$\{0, 1\}^* = \{(), (0), (1), (0, 0), (0, 1), (1, 0), (1, 1), (0, 0, 0), \dots\}$$

We already know that the number of binary strings of length n is 2^n, but we can see that using the string lemma as well:

$$\begin{aligned}
[x^n]\Phi_{\{0,1\}^*}(x) &= [x^n]\frac{1}{1-\Phi_{\{0,1\}}(x)} && \text{(string lemma)}\\
&= [x^n]\frac{1}{1-2x} && \text{(gen. series of } \{0,1\} \text{ w.r.t. length is } 2x)\\
&= 2^n && \text{(geometric series)}
\end{aligned}$$

We'll use generating series to count binary strings which have various properties, such as the number of binary strings of length 10 not containing '101'.

We can **concatenate** binary strings: if $\sigma = a_1a_2\dots a_m$ and $\tau = b_1b_2\dots b_n$ are binary strings, then the concatenation $\sigma\tau$ is the binary string $a_1a_2\dots a_mb_1b_2\dots b_n$. And σ^k denotes the k-fold concatenation of σ with itself: $\sigma^k = \underbrace{\sigma\sigma\dots\sigma}_{k}$.

We say that σ is a **substring** of τ if there exist binary strings γ_1, γ_2 such that

$$\gamma_1\sigma\gamma_2 = \tau$$

Definition 10 (Concatenation product, Definition 3.3 in the notes)**.** If S and T are sets of binary strings, then

$$ST = \{\sigma\tau : \sigma \in S, \tau \in T\}$$

and $S^k = \underbrace{SS\dots S}_{k}$.

Example 20.

$$\{0,1\}\{\epsilon, 00, 11\} = \{0, 000, 011, 1, 100, 111\}$$
$$\{00, 01\}^2 = \{0000, 0001, 0100, 0101\}$$

3.1 Regular expressions (§3.1)

Definition 11 (Regular expression, Definition 3.2 in the notes)**.** A **regular expression** is defined recursively as any of the following:

- ϵ, 0, or 1;
- the expression $R \smile S$ where R and S are regular expressions;
- the expression RS where R and S are regular expressions (with $R^k = \underbrace{RR\dots R}_{k}$ for any $k \in \mathbb{N}$); or
- the expression R^*, where R is a regular expression.

Example 21. The following are all regular expressions:

- 010
- $010 \smile 01$
- $(010 \smile 01)^*$
- $(11)(010 \smile 01)^*(\epsilon \smile 0*)$
- $(00)^5 \smile ((11)(010 \smile 01)^*(\epsilon \smile 0*))$

A regular expression R **produces** a set $\mathcal{R}$ of binary strings.

Example 22.

$R_1 = \epsilon \smile 0 \smile 1$	produces	$\mathcal{R}_1\{\epsilon, 0, 1\}$
$R_2 = (01)(0 \smile 11)$	produces	$\mathcal{R}_2 = \{010, 0111\}$
$R_3 = (00 \smile 11)^*$	produces	$\mathcal{R}_3 = \{\epsilon, 00, 11, 0000, 0011, 1100, 1111, 000000, \dots\}$

Definition 12 (Rational languages, Definition 3.5 in the notes)**.** We define **production** recursively:

- The regular expressions ϵ, 0, and 1 produce the sets $\{\epsilon\}, \{0\}, \{1\}$ respectively.
- If R produces $\mathcal{R}$ and S produces $\mathcal{S}$, then
  - $R \smile S$ produces $\mathcal{R} \cup \mathcal{S}$ (set union)
  - RS produces $\mathcal{RS}$ (concatenation product)
  - R^* produces $\mathcal{R}^* = \cup_{k \geq 0} \mathcal{R}^k$ (concatenation powers)

Example 23.

0^*	produces	$\{\epsilon, 0, 00, 00, 000, \dots\}$
$(0 \smile 1)^*$	produces	{ all binary strings }
$(11)0^*$	produces	$\{11, 110, 1100, 11000, 11000, 110000, \dots\}$
$(00 \smile 000)^*$	produces	$\{\epsilon, 00, 000, 0000, 00000, \dots\}$
$(0 \smile 00)^*$	produces	the same as 0^*

Definition 13. If $\mathcal{R}$ is a set of strings that can be produced by a regular expression R, then we say that $\mathcal{R}$ is a **rational language**.

Example 24.

- $\{1, 110, 011, \epsilon, 0\}$ is a rational language since can be produced by $1 \smile 110 \smile 011 \smile \epsilon \smile 0$
- $\{\underbrace{1010\dots10}_{m}\underbrace{00\dots0}_{n} : m, n \geq 1\}$ is a rational language since it can be produced by $10(10)^*00^*$
- $\{\epsilon, 01, 0011, 000111, \dots, \underbrace{00\dots0}_{n}\underbrace{11\dots1}_{n}\}$ is not a rational language. (Why?)

3.2 Unambiguous expressions (§3.2)

Example 25.

- $(0 \smile 01)(0 \smile 10)$ produces $\{0, 01\}\{0, 10\} = \{00, 010, 010, 0110\} = \{00, 010, 0110\}$. Notice that 010 is produced twice.
- $(0 \smile 1)^*$ produces every binary string exactly once.
- $(0 \smile 1 \smile 01)^*$ produces every binary string but some are produced more than once.

Definition 14 (Unambiguous expressions, Definition 3.8 in the notes)**.** A regular expression R is **unambiguous** if every string in the language $\mathcal{R}$ produced by R is produced in exactly one way by R. Otherwise, R is called **ambiguous**.

Lemma 5 (Unambiguous expressions, Lemma 3.9 in the notes)**.**

- *The regular expressions ϵ, 0, and 1 are unambiguous expressions.*
- *If R and S are unambiguous expressions that produce $\mathcal{R}$ and $\mathcal{S}$ respectively, then*
  - *$R \smile S$ is unambiguous if and only if $\mathcal{R} \cap \mathcal{S} = \emptyset$, in other words they are disjoint*
  - *RS is unambiguous if and only if there is a bijection between $\mathcal{RS}$ and $\mathcal{R} \times \mathcal{S}$. In other words, for every string $\alpha \in \mathcal{RS}$, there is exactly one way to write $\alpha = \rho\sigma$ with $\rho \in \mathcal{R}$ and $\sigma \in \mathcal{S}$*
  - *R^* is unambiguous if and only if each of the concatenation products R^k is unambiguous and all the R^k are disjoint.*

Example 26. Which of the following are unambiguous? Why or why not?

- 010: unambiguous
- $(0 \smile 1)^*$: unambiguous. Each string can be obtained in exactly one way by concatenating strings in $\{0, 1\}$.
- $(0 \smile 1)^*(0 \smile 1)^*$: ambiguous. Many strings can be produced in multiple ways. For example, 00 could be produced as $(00)\epsilon$ or $(0)(0)$ or $\epsilon(00)$.

Example 27. $0^*(10^*)^*$ is an unambiguous expression that produces the set of all binary strings.

To see this, notice that a string produced by $0^*(10^*)^*$ has the form

$$\underbrace{0\ldots0}_{m_0}(1\underbrace{0\ldots0}_{m_1})(1\underbrace{0\ldots0}_{m_2})\ldots(1\underbrace{0\ldots0}_{m_k})$$

for $k \geq 0$ and $m_0, m_1, \ldots, m_k \geq 0$.

Every binary string $\sigma \in \{0, 1\}^*$ can be written in this way: k is the number of 1's, and m_i is the number of zeroes in the block immediately before the $(i + 1)$th 1.

Example 28. $1^*(01^*)^*$ is also an unambiguous expression that produces the set of all binary strings.

Example 29. Give an unambiguous expression that produces the set of all binary strings where every 0 is followed by at least three 1's.

$$1^*(01111^*)^*$$

3.3 Translation into generating series (§3.2.1)

Example 30. How many strings of length 10 are there where the length of every block of 1's is even?

Let $\mathcal{R}$ be the set of such strings. An unambiguous expression producing $\mathcal{R}$ is

$$R = (0 \smile 11)^*$$

Since R is unambiguous, there's a bijection between $\{0, 11\}^*$ and $\mathcal{R}$. So

$$\begin{aligned}
\text{\# of strings in } \mathcal{R} \text{ of length 10} &= \text{\# of strings in } \{0,11\}^* \text{ of length 10} \\
&= [x^{10}]\Phi_{\{0,11\}^*}(x) && \text{(with weight function = length)} \\
&= [x^{10}]\frac{1}{1-\Phi_{\{0,11\}}(x)} && \text{(string lemma)} \\
&= [x^{10}[\frac{1}{1-(x+x^2)}
\end{aligned}$$

We went from unambiguous expression $R = (0 \smile 11)^*$ to the equivalent set $\{0, 11\}^*$ so we could count strings produced by R using generating series $\Phi_{\{0,11\}^*}(x) = \frac{1}{1-x-x^2}$. We say that regular expression R **leads to** the rational function $\frac{1}{1-x-x^2}$.

Definition 15 (Leads to, Definition 3.11 in the notes)**.** A regular expression **leads to** a rational function, defined recursively as follows.

- Regular expressions ϵ, 0, and 1 lead to formal power series 1, x, and x
- If R and S are regular expressions that lead to $f(x)$ and $g(x)$, then
  - $R \smile S$ leads to $f(x) + g(x)$
  - RS leads to $f(x) \cdot g(x)$
  - R^* leads to $\frac{1}{1-f(x)}$.

Example 31. 1100 leads to $x^1 \cdot x^1 \cdot x^1 \cdot x^1 = x^4$
$11 \smile 000$ leads to $x^2 + x^3$
$0 * (11)(11 \smile 000)^*$ leads to $\frac{1}{1-x}x^2\frac{1}{1-x^2-x^3}$

Theorem 9 (Theorem 3.13 in the notes)**.** *Let R be a regular expressions that unambiguously produces the language $\mathcal{R}$. Also suppose that R leads to $f(x)$. Then the generating series for $\mathcal{R}$ with respect to length is $f(x)$, i.e., $\Phi_{\mathcal{R}}(x) = f(x)$.*

3.4 Block decompositions (§3.2.2)

Definition 16 (Blocks of a string, Definition 3.15 in the text)**.** A **block** of a binary string s is a nonempty maximal substring of equal bits.

Example 32. 00011010000 has 5 blocks: $000, 11, 0, 1, 0000$.

Proposition 4 (Block decomposition, Proposition 3.17 in the notes)**.** *The regular expression* $0^*(11^*00^*)^*1^*$ *is unambiguous and produces the set of all binary strings. Same for* $1^*(00^*11^*)^*0^*$.

Example 33. Show how each of the above block decompositions can produce the string 00011010000.

$(0000)(110)(10000)()$
$()(000011)(01)(0000)$

The regular expressions in the block decomposition work by splitting the string into blocks; the "forced" 1 and 0 inside the parenthesis act as block delimiters.

Example 34. Give a regular expression for the set of all binary strings where every block of 1's has even length.

Start from the block decomposition $1^*(00^*11^*)^*0^*$ and force all blocks of 1 to be even length:

$$R = \underbrace{(11)^*}_{\text{even-sized block of 1s}} (00^* \underbrace{11(11)^*}_{\text{even-sized block of 1s}})^*0^*$$

Example 35. Give a regular expression for the set of all binary strings not containing 0000 as a substring. How many are there of length 15?

Start from the block decomposition $0^*(11^*00^*)^*1^*$ and restrict blocks of 0's to be of length 3 or less.

$$R = (\epsilon \smile 0 \smile 00 \smile 000)(11^*(0 \smile 00 \smile 000))^*1^*$$

This regular expression leads to the generating series

$$\Phi(x) = \underbrace{(1 + x + x^2 + x^3)}_{(\epsilon\smile 0\smile 00\smile 000)} \frac{1}{1 - x\underbrace{\frac{1}{1-x}}_{11^*}\underbrace{(x + x^2 + x^3)}_{0\smile 00\smile 000}} \underbrace{\frac{1}{1-x}}_{1^*}$$

So we want $[x^{15}]\Phi(x)$

Example 36. How many strings of length 15 are there where every block has length 1 or 2?

Start from the block decomposition $1^*(00^*11^*)^*0^*$ and force all blocks to be of length 1 or 2.

$$R = (\epsilon \smile 1 \smile 11)\left((0 \smile 00)(1 \smile 11)\right)^* (\epsilon \smile 0 \smile 00)$$

This leads to generating series

$$f(x) = (1+x+x^2)\frac{1}{1-(x+x^2)(x+x^2)}(1+x+x^2) \tag{1}$$
$$= \frac{1+x+x^2}{1-x-x^2} \qquad \text{(difference of squares in denominator)}$$

So the number of strings of length 15 with all blocks of length 1 or 2 is $[x^{15}]\frac{1+x+x^2}{1-x-x^2}$. (This happens to be 2 times the 15th Fibonacci number.)

Example 37. Find the generating series for the set of all strings not containing 011 as a substring.

Start from the block decomposition $1^*(00^*11^*)^*0^*$ and restrict to at most one 1 after any block of 0's.

$$R = 1^*(00^*1)^*0^*$$

This leads to generating series

$$\begin{aligned}
f(x) &= \frac{1}{1-x}\frac{1}{1-\frac{x}{1-x}x}\frac{1}{1-x} \\
&= \frac{1}{1-x-x^2}\frac{1}{1-x} \\
&= \frac{1}{(1-x-x^2)(1-x)} \\
&= \frac{1}{1-x-x^2+x+x^2+x^3} \\
&= \frac{1}{1+x^3}
\end{aligned}$$

3.5 Prefix decompositions (§3.2.3)

Another unambiguous expression for the set of all binary strings is $(0^*1)^*0^*$. We can see this as chopping a string into pieces after each 1. For example, chop

$$1100001101 \quad \text{as} \quad 1.1.00001.1.01$$

This is an example of a **prefix decomposition** which has the form A^*B. (When using decompositions of this form, need to confirm it's unambiguous. Usually can do this by checking that A and B are unambiguous, that there's at most one way for a string to start with an initial segment produced by A, and if it doesn't start with a segment produced by A then it's produced by B.)

A **postfix decomposition** has the form AB^*. For example, the set of all binary strings is also produced by $0^*(10^*)$.

3.6 Recursive decompositions (§3.3)

Now we'll define something more powerful than regular expressions: **recursive expressions**, which can reference itself.

Example 38. A recursive expression that produces the set of all binary strings is

$$S = \epsilon \smile S(0 \smile 1)$$

This is unambiguous because every non-empty binary string can be uniquely expressed as another binary string with a 0 or a 1 appended.

If S leads to generating series $f(x)$, then

$$S = \epsilon \smile S(0 \smile 1) \quad \text{leads to} \quad f(x) = x^0 + f(x)(x^1 + x^1)$$

Solving $f(x) = 1 + f(x)x^2$ for $f(x)$ yields $f(x)(1 - 2x) = 1$ and thus $f(x) = 1/(1 - 2x)$.

A **recursive decomposition** of a set $\mathcal{S}$ describes S in terms of itself using the language of regular expressions together with the symbol S which produces set $\mathcal{S}$.

A recursive decomposition for S is **unambiguous** if each side of the equation produces each string at most once.

Example 39. $S = 1S1 \smile 0$ describes $\mathcal{S} = \{0, 101, 11011, \dots\}$. A string $\sigma \in S$ is either 0 or it is $1\sigma 1$ for some $\sigma \in S$. It is unambiguous because each string in S is either 0 or has the form $1\sigma 1$ for some unique $\sigma \in S$.

$S = \epsilon \smile 0 \smile 1S1$ unambiguously describes $\mathcal{S} = \{\epsilon, 0, 11, 101, 1111, 11011, 111111, 1110111, \dots\}$.

$S = 0 \smile 00 \smile 0S$ describes $\mathcal{S} = \{0, 00, 000, 0000, \dots\}$ but is ambiguous.

An unambiguous recursive decomposition leads to the generating series for the corresponding set.

Example 40. $S = \epsilon \smile 0 \smile 1S1$ gives

$$\begin{aligned}
\Phi_S(x) &= \Phi_{\{\epsilon\}}(x) + \Phi_{\{0\}}(x) + \Phi_{\{1\}}(x)\Phi_S(x)\Phi_{\{1\}}(x) + \\
\Phi_S(x) &= 1 + x + x\Phi_S(x)x \\
\Phi_S(x)(1 - x^2) &= 1 + x \\
\Phi_S(x) &= \frac{1 + x}{1 - x^2} = \frac{1}{1 - x}
\end{aligned}$$

Recursive decompositions allow us to produce sets that are not rational languages, i.e., that cannot be produced by regular expressions.

Example 41. Give a recursive decomposition for the set $\mathcal{S} = \{0^n 1^n : n \in \mathbb{N}\}$.

$$S = \epsilon \smile 0S1$$

3.7 Excluded substrings (§3.3.1)

Example 42. Find the generating series for the set of strings without 10100 as a substring.

Let A be the set of strings avoid 10100, and let B be the set of strings that have exactly 1 occurrence of 10100, at the very end.

Consider the set $A \cup B$. Other than the empty string, every string in $A \cup B$ either ends with a 0 or 1. If it ends with a 1, then the prefix (excluding the last 1) must already avoid 10100, so the prefix is in A. If it ends with a 0, then either the last 5 bits were 10100 or they weren't. If they were, then the string was in B, so no other occurrence of 10100 appears, so the prefix is in A. If they last 5 bits weren't 10100, then the prefix is also in A. In other words,

$$A \cup B = \epsilon \smile A(0 \smile 1)$$

This leads to the formula on generating series

$$A(x) + B(x) = 1 + 2xA(x)$$

We still need another connection between A and B to solve the question.

We observe that every string in B can be uniquely obtained by appending 10100 to a string in A, and that appending 10100 to a string in A always gives a string in B. To justify the first part: if $b \in B$, then it is of the form $b = \sigma 10100$ where σ does not contain 10100, which means $\sigma \in A$. To justify the second part: suppose $\sigma \in A$. Is $\sigma 10100 \in B$? The only problem would be if $\sigma 10100$ contains a 2 copies of 10100. We know σ doesn't, what about the overlap? This is not possible for 10100. So we can write $B = A10100$ and thus

$$B(x) = x^5 A(x)$$

Substitute this into the first equation to get

$$A(x) + x^5 A(x) = 1 + 2xA(x)$$

which implies

$$A(x) = \frac{1}{1 - 2x + x^5}$$

Example 43. Find the generating series for

$$\mathcal{S} = \{\text{strings without 101 as a substring}\}$$

Define

$$\mathcal{A} = \{\text{strings with exactly 1 occurrence of 101, appearing in the last 3 bits}\}$$

$$\mathcal{B} = \{\text{strings with exactly 2 occurrences of 101, appearing in the last 5 bits}\}$$

We have 2 cases here because of the potential overlap: 10101.

We can derive:

- $\epsilon \smile S(0 \smile 1) = S \smile A$ leads to $1 + 2x\Phi_S(x) = \Phi_S(x) + \Phi_A(x)$
- $S101 = A \smile B$ leads to $x^3\Phi_S(x) = \Phi_A(x) + \Phi_B(x)$
- $A01 = B$ leads to $x^2\Phi_A(x) = \Phi_B(x)$

Putting all these equations together and solving for $\Phi_S(x)$ we get

$$\Phi_S(x) = \frac{1 + x^2}{1 - 2x + x^2 - x^3}$$

4 Partial Fractions and Recurrence Relations (§4)

Let's try to extract coefficients when the generating series has a multi-term denominator using partial fractions.

Example 44.

$$\begin{aligned}
[x^7]\frac{1+7x}{1-x-6x^2} &= [x^7]\frac{1+7x}{(1+2x)(1-3x)} && \text{(factoring)}\\
&= [x^7]\frac{-1}{1+2x}+\frac{2}{1-3x} && \text{(partial fractions)}\\
&= [x^7]\left(-1\sum_{n\geq 0}(-2x)^n + 2\sum_{n\geq 0}(3x)^n\right) && \text{(geometric series)}\\
&= [x^7]\sum_{n\geq 0}(-1(-2)^n + 2\cdot 3^n)x^n\\
&= 2\cdot 3^7 + 2^7
\end{aligned}$$

Theorem 10 (Partial fractions (simple version), Theorem 4.12 in the notes). *Let*

$$G(x) = \frac{P(x)}{(1-\lambda_1 x)(1-\lambda_2 x)\dots(1-\lambda_s x)}$$

where P is a polynomial of degree less than s and $\lambda_i \in \mathbb{C}$ are distinct. Then there exist $C_1, C_2, \dots, C_s \in \mathbb{C}$ such that

$$G(x) = \frac{C_1}{1-\lambda_1 x} + \frac{C_2}{1-\lambda_2 x}\cdots\frac{C_s}{1-\lambda_s x}$$

To find these C_i, cross-multiply and equate coefficients.

Example 45.

$$\frac{1+7x}{(1+2x)(1-3x)} = \frac{C_1}{1+2x} + \frac{C_2}{1-3x}$$

implies

$$1+7x = C_1(1-3x) + C_2(1+2x) = (C_1+C_2) + (-3C_1+2C_2)x$$

so $C_1 + C_2 = 1$ and $-3C_2 + 2C_2 = 7$, yielding $C_2 = 2$ and $C_1 = -1$.

Theorem 11 (Partial fractions, Theorem 4.12 in the notes). *Let*

$$G(x) = \frac{P(x)}{Q(x)} = \frac{P(x)}{(1-\lambda_1 x)^{d_1}(1-\lambda_2 x)^{d_2}\dots(1-\lambda_s x)^{d_s}}$$

where $deg(P) < deg(Q)$, the $\lambda_i \in \mathbb{C}$ are distinct, and $d_i \geq 1$. Then there exist $C_1^{(1)}, C_1^{(2)}, \dots, C_1^{(d_1)}$, $C_2^{(1)}, \dots, C_2^{(d_2)}, \dots, C_s^{(1)}, \dots, C_s^{(d_s)} \in \mathbb{C}$ such that

$$G(x) = \sum_{i=1}^{s}\sum_{j=1}^{d_i}\frac{C_i^{(j)}}{(1-\lambda_i x)^j}$$

Example 46.

$$\begin{aligned}
G(x) &= \frac{2+5x}{1-3x^2-2x^3} \\
&= \frac{2+5x}{(1+x)^2(1-2x)} && \text{(factoring)} \\
&= \frac{A}{1+x} + \frac{B}{(1+x)^2} + \frac{C}{1-2x} && \text{(partial fractions)} \\
2+5x &= A(1+x)(1-2x) + B(1-2x) + C(1+x)^2 && \text{(clear denominator)} \\
2+5x &= A(1+x-2x-2x^2) + B(1-2x) + C(1+2x+x^2) \\
2+5x+0x^2 &= (A+B+C) + (-A-2B+2C)x + (-2A+0B+C)x^2
\end{aligned}$$

which leads to the system of linear equations

$$\begin{aligned}
2 &= A+B+C \\
5 &= -A-2B+2C \\
0 &= -2A+0B+C
\end{aligned}$$

which is satisfied by $(A,B,C) = (1,-1,2)$ so

$$\frac{2+5x}{1-3x^2-2x^3} = \frac{1}{1+x} - \frac{1}{(1+x)^2} + \frac{2}{1-2x}$$

Now we can use geometric series and the negative binomial theorem to get a single sum:

$$\begin{aligned}
&= \sum_{n\geq 0}(-x)^n - \sum_{n\geq 0}\binom{n+2-1}{2-1}(-x)^n + 2\sum_{n\geq 0}(2x)^n \\
&= \sum_{n\geq 0}\left((-1)^n - (n+1)(-1)^n + 2\cdot 2^n\right)x^n \\
&= \sum_{n\geq 0}\left(2^{n+1} - n(-1)^n\right)x^n
\end{aligned}$$

Summary: to analyze coefficients of a generating series like

$$G(x) = \frac{2+5x}{1-3x^2-2x^3}$$

our process will be:

1. Factor the denominator: $G(x) = \frac{2+5x}{(1+x)^2(1-2x)}$
2. Write it as a sum of negative powers of polynomials using partial fractions: $G(x) = \frac{A}{1+x} + \frac{B}{(1+x)^2} + \frac{C}{1-2x}$
3. Solve for A, B, C by clearing the denominator: $(A,B,C) = (1,-1,2)$.
4. Use geometric series and negative binomial theorem to write $G(x)$ as a single sum: $G(x) = \sum_{n\geq 0}\left(2^{n+1} - n(-1)^n\right)x^n$

4.1 Recurrence relations

We previously saw that the generating series for the number of compositions with only odd parts is:

$$G(x) = \frac{1-x^2}{1-x-x^2} = \sum_{n\geq 0} g_n x^n = g_0 + g_1 x + g_2 x^2 + \dots$$

Let's try to find a recurrence relation for g_n, the number of compositions of n with only odd parts.

Multiplying both sides by the denominator, we have

$$\begin{aligned} 1 + 0x - x^2 &= (1-x-x^2)(g_0 + g_1 x + g_2 x^2 + \dots) \\ &= g_0 + (g_1 - g_0)x + (g_2 - g_1 - g_0)x^2 + (g_3 - g_2 - g_1)x^3 + \dots) \end{aligned}$$

which implies

$$\begin{aligned} 1 &= g_0 \\ 0 &= g_1 - g_0 \\ -1 &= g_2 - g_1 - g_0 \\ 0 &= g_n - g_{n-1} - g_{n-2} \text{ for } n \geq 3 \end{aligned}$$

Simplifying yields $g_0 = 1, g_1 = 1, g_2 = 1, g_n = g_{n-1} + g_{n-2}$ for $n \geq 3$. (Which you might recognize as the Fibonacci sequence.)

Speaking of the Fibonacci sequence, let's consider it as a separate example. Recall that the Fibonacci sequence is defined by:

$$f_0 = 1 \quad , \quad f_1 = 1 \quad , \quad f_n = f_{n-1} + f_{n-2} \text{ for } n \geq 2$$

Let $F(x) = \sum_{n\geq 0} f_n x^n$ be the generating series for the Fibonacci sequence.

The recurrence relation for the Fibonacci sequence is, for $n \geq 2$:

$$\begin{aligned} f_n &= f_{n-1} + f_{n-2} \\ 0 &= f_n - f_{n-1} - f_{n-2} && \text{(rearranging)} \\ &= [x^n]F(x) - [x^{n-1}]F(x) = [x^{n-2}]F(x) && \text{(since } f_i = [x^i]F(x)\text{)} \\ &= [x^n]F(x) - [x^n]xF(x) = [x^n]x^2F(x) && \text{(rules of coefficient extraction)} \\ &= [x^n](1-x-x^2)F(x) \end{aligned}$$

So, for $n \geq 2$, $[x^n](1-x-x^2)F(x) = 0$. This means that $(1-x-x^2)F(x)$ must be of the form $C_1 + C_2 x + 0x^2 + 0x^3 + \dots$ for some $C_1, C_2 \in \mathbb{C}$.

In other words,

$$(1-x-x^2)F(x) = C_1 + C_2 x \quad \text{or equivalently} \quad F(x) = \frac{C_1 + C_2 x}{1-x-x^2}$$

To get a closed form for the coefficients of $F(x)$, we'll adapt our technique summarized at the end of the previous section.

We are starting from

$$F(x) = \frac{C_1 + C_2 x}{1 - x - x^2}$$

Factor the denominator:

$$F(x) = \frac{C_1 + C_2 x}{(1 - \alpha x)(1 - \beta x)}$$

where $\alpha, \beta = \frac{1 \pm \sqrt{5}}{2}$.

Write it as a sum using partial fractions:

$$F(x) = \frac{A}{1 - \alpha x} + \frac{B}{1 - \beta x}$$

Apply geometric series:

$$F(x) = \sum_{n \geq 0} \left(A \cdot \alpha^n + B \cdot \beta^n\right) x^n$$

So

$$f_n = [x^n] F(x) = A \cdot \alpha^n + B \cdot \beta^n$$

Substitute in for our initial conditions $f_0 = 1, f_1 = 1$ to get

$$f_0 = 1 = A\alpha^0 + B\beta^0 \implies A + B = 1$$
$$f_1 = 1 = A\alpha^1 + B\beta^1 \implies A\frac{1+\sqrt{5}}{2} + B\frac{1-\sqrt{5}}{2} = 1$$

Solve the system of 2 linear equations in two unknowns to get:

$$A = \frac{5 + \sqrt{5}}{10} \quad , \quad B = \frac{5 - \sqrt{5}}{10}$$

We can conclude:

$$f_n = \frac{5 + \sqrt{5}}{10} \left(\frac{1 + \sqrt{5}}{2}\right)^n + \frac{5 - \sqrt{5}}{10} \left(\frac{1 - \sqrt{5}}{2}\right)^n$$

The big picture was:

1. Start from recurrence: $f_n - f_{n-1} - f_{n-2} = 0$ for $n \geq 2$
2. Write a rational expression and factor the denominator:

$$F(x) = \frac{C_1 + C_2 x}{1 - x - x^2} = \frac{C_1 + C_2 x}{(1 - \alpha x)(1 - \beta x)}$$

3. Apply partial fractions to write expression as a sum:
$$F(x) = \frac{A}{1-\alpha x} + \frac{B}{1-\beta x}$$

4. Apply geometric and negative binomial theorem to obtain coefficient formula:
$$f_n = [x^n]F(x) = A\alpha^n + B\beta^n$$

5. Substitute initial conditions $f_0 = 1, f_1 = 1$ to solve for A and B.

The following theorem justifies this process:

Theorem 12 (Combination of Theorem 4.8 and 4.14)**.** *Let $c_1, \dots, c_k, \lambda_1, \dots, \lambda_s \in \mathbb{C}$ with the λ_i distinct, such that*
$$\underbrace{1 + c_1x + c_2x^2 + \cdots + c_kx^k}_{\text{“characteristic polynomial”}} = (1-\lambda_1x)^{d_1}(1-\lambda_2x)^{d_2}\dots(1-\lambda_sx)^{d_s}$$
If $a_0, a_1, \dots$ is the sequence satisfying the recurrence relation
$$\underbrace{a_n + c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k} = 0}_{\text{“homogeneous linear recurrence”}}$$
for all $n \geq k$ then
$$a_n = p_1(n)\lambda_1^n + p_2(n)\lambda_2^n + \cdots + p_s(n)\lambda_s^n$$
where each p_i is a polynomial of degree less than d_i.

Example 47. Suppose a_n is a sequence given by $(a_0, a_1, a_2) = (0, -5, -1)$ and $a_n = 3a_{n-2} - 2a_{n-3}$ for $n \geq 3$. Give a formula for a_n as a function of n.

Write the recurrence as $a_n - 3a_{n-2} + 2a_{n-3} = 0$. The characteristic polynomial of the recurrence relation is
$$1 - 3x^2 + 2x^3 \text{ which factors as } (1-x)^2(1+2x)$$
So by the theorem we can write
$$a_n = (An + B)1^n + C(-2)^n$$
Substituting in our initial conditions a_0, a_1, a_2 and solving for A, B, C yields $(A, B, C) = (-2, -1, 1)$. So
$$a_n = -2n - 1 + (-2)^n$$

4.2 Quadratic recurrence relations (§4.4)

We previously looked at generating series that satisfied linear recurrence relations and could be written as a rational function
$$G(x) = \frac{P(x)}{Q(x)}$$

or equivalently as a linear equation in G:

$$Q(x)G(x) - P(x) = 0$$

Definition 17. A sequence $(g_0, g_1, \dots)$ satisfies a *quadratic recurrence* if its generating series $G(x)$ satisfies a quadratic equation

$$A(x)G(x)^2 + B(x)G(x) + C(x) = 0$$

where $A(x), B(x), C(x)$ are power series in x.

There are two solutions using the quadratic formula:

$$\left.\begin{matrix} G_1(x) \\ G_2(x) \end{matrix}\right\} = \frac{-B(x) \pm \sqrt{B(x)^2 - 4A(x)C(x)}}{2A(x)}$$

One of these (the ones with non-negative coefficients and exponents) will be the generating series.

We can generalize the binomial coefficient and series to complex numbers.

Definition 18 (Complex binomial coefficient (Definition 4.20))**.** Let $\alpha \in \mathbb{C}$ and $k \in \mathbb{N}$. The *kth binomial coefficient of* α is

$$\binom{\alpha}{k} = \frac{\alpha(\alpha-1)\cdot(\alpha-k+1)}{k!}$$

Theorem 13 (Complex binomial series (Theorem 4.21 in the notes))**.** *For any* $\alpha \in \mathbb{C}$,

$$(1+x)^\alpha = \sum_{k \geq 0} \binom{\alpha}{k} x^k$$

Here's one special case:

Proposition 5.

$$\sqrt{1-4x} = 1 - 2\sum_{k\geq 1} \frac{1}{k}\binom{2k-2}{k-1} x^k = 1 - 2\sum_{k \geq 0} \frac{1}{k+1}\binom{2k}{k} x^{k+1}$$

Proof. By the complex binomial series theorem,

$$\sqrt{1-4x} = \sum_{k\geq 0} \binom{1/2}{k} (-1)^k 4^k x^k$$

For $k = 0$, this simplifies to $1x^0$.

For $k \geq 1$:

$$\begin{aligned}
\binom{1/2}{k}(-1)^k 4^k &= \frac{(1/2)(-1/2)(-3/2)\dots(1/2-k+1)}{k!}(-1)^k 4^k \\
&= (-1)4^k \frac{1}{k!}(1/2)(1/2)(3/2)\dots(k-3/2) && \text{(factoring out the } -1) \\
&= -2^k \frac{1}{k!}(1)(1)(3)\dots(2k-3) && \text{(multiplying through by two } k \text{ times)} \\
&= -2^k \frac{1}{k!}(1)(3)\dots(2k-3)\frac{(k-1)!}{(k-1)!} \\
&= -2\frac{1}{k!}(1)(3)\dots(2k-3)\frac{(2)(4)\dots(2k-2)}{(k-1)!} \\
& && \text{(multiplying through by two } k-1 \text{ times)} \\
&= -2\frac{1}{k}\frac{(1)(3)\dots(2k-3)}{(k-1)!}\frac{(2)(4)\dots(2k-2)}{(k-1)!} \\
&= -\frac{2}{k}\frac{(2k-2)!}{(k-1)!(k-1)!} \\
&= -\frac{2}{k}\binom{2k-2}{k-1}
\end{aligned}$$

□

4.3 Catalan numbers (§4.4.2)

The nth Catalan number is

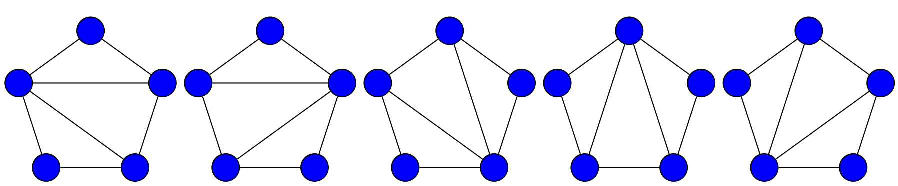
$$C_n = \frac{1}{n+1}\binom{2n}{n} \quad \text{or equivalently} \quad C_n = \binom{2n}{n} - \binom{2n}{n+1}$$

The first few (starting from $n = 0$) are:

$$1, 1, 2, 5, 14, 42, 132, \dots$$

Catalan numbers pop up in many places:

- the number of full binary trees with $n + 1$ leaves
- the number of ways a convex polygon with $n + 2$ sides can be cut into triangles by connecting vertices with non-crossing line segments



- the number of well-formed parentheses

Example 48. A *well-formed parenthesization* is a sequence of n opening and n closing parentheses which "match". Here are all the WFPs of size 3:

$$()()() \quad ()(()) \quad (())() \quad (()()) \quad ((()))$$

Find the generating series $W(x) = \sum_{n\geq 0} w_n x^n$ where w_n is the number of WFPs of size n.

The empty string ϵ is a WFP of size 0. We can build a WFP of size $n+1$ by taking a WFP of size n and putting parentheses around it, or by appending a WFP to it. This gives the unambiguous recursive decomposition:

$$\mathcal{W} = \epsilon \smile (\mathcal{W})\mathcal{W}$$

So this leads to

$$W(x) = 1 + xW(x)W(x) = 1 + xW(x)^2$$

where the single x comes from the opening/closing parenthesis pair we added. This is the quadratic equation

$$xW(x)^2 - W(x) + 1 =$$

which is solved by

$$\begin{aligned}
\left.\begin{matrix} W_1(x) \\ W_2(x) \end{matrix}\right\} &= \frac{1 \pm \sqrt{1-4x}}{2x} && \text{(quadratic formula)} \\
&= \frac{1}{2x} \pm \frac{1}{2x}\left(1 - 2\sum_{n\geq 0} \frac{1}{n+1}\binom{2n}{n} x^{n+1}\right) && \text{(proposition above)} \\
W_2 &= \frac{1}{2x} - \frac{1}{2x} + \sum_{n\geq 0} \frac{1}{n+1}\binom{2n}{n} x^n \\
& && \text{(taking minus version and multiplying through by } 1/2x\text{)} \\
&= \sum_{n\geq 0} \frac{1}{n+1}\binom{2n}{n} x^n \\
&= \sum_{n\geq 0} C_n x^n
\end{aligned}$$

Since the generating series W is not a rational function, it follows that the set of WFPs is not a rational language and thus cannot be generated by a regular expression.