MATH239 Final Exam Guide

Richard Wu

Contents

1 Combinatorial Analysis

1.1 Union, Catersian Product and Power

Union $A \cup B$ If A and B are disjoint, then

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

Cartesian product $A \times B$ Note that

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

and the following holds

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

Cartesian power A^k As an extension to the Cartesian product

$$A^k = \{(a_1, \ldots, a_k) : a_i \in A, \forall i\}$$

the following holds

$$|A^k| = |A|^k$$

Note $A^0 = \{()\}$ (set with one empty tuple).

1.2 Binomial Coefficient

Given n objects, you want to choose/select k objects without replacement. Order does not matter. The number of ways to do this is

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

1.3 Bijections

A bijection $f : S \to T$ holds if

f **is injective** Also called **1-1**, every element in S can only map to one element in T (vertical line test). Formally

$$f(x_1) = f(x_2) \iff x_1 = x_2$$

f **is surjective** Also called **onto**, every element in T is mapped to by an element in S (the codomain T is equivalent to the range of f). Formally

$$\forall t \in T, \exists s \in S : f(s) = t$$

The trick to show a bijection between two sets is to show for $f : S \to T$ a unique way of mapping each element $s \in S$ to an element $t \in T$, and similar $f^{-1} : T \to S$ (show **1-1** both ways).

1.4 Combinatorial Proofs

To show that an equation with combinatorial arguments holds, one can formulate a set of objects and show two ways of counting: each way of counting corresponds to one side of each equation.

One can proof combinatorial equalities by algebraic manipulation if permitted.

1.5 Series Identities

The Binomial Series identities can be derived using Taylor Series expansion about $x = 0$

Binomial Series

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}$$

where $n \in \mathbb{N}$

Negative Binomial Series

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

Geometric Series identities are useful in simplifying generating series

Finite Geometric Series

$$a + ar + ar^2 + \ldots + ar^{n-1} = \frac{a(1-r^n)}{1-r}$$

Infinite Geometric Series

$$a + ar + ar^2 + \ldots = a\sum_{i\geq 0} r^i = \frac{a}{1-r}$$

1.6 Compositions

A composition of n is an *ordered* sequence $(\alpha_1, \ldots, \alpha_k)$ of k positive integers where

$$\alpha_1 + \ldots + \alpha_k = n$$

1.7 Generating Series

Generating series lets us abstract counting the number of ways to choose a **configuration** $\sigma \in S$ into a nice, algebraic form.

A generating series for a set of configurations S follows the form

$$\Phi_S(x) = \sum_{\sigma \in S} x^{w(\sigma)}$$

where $w(\sigma)$ is the **weight function** for each configuration σ. Think of the weight function as a way of quantifying the value we're interested in each configuration For a composition of k parts (σ), we are interested in the composition value or the sum of its parts. Thus if $\sigma = (\alpha_1, \ldots, \alpha_k)$

$$w(\sigma) = \alpha_1 + \ldots + \alpha_k$$

Some useful identities arise when we set $x = 1$:

$\Phi_S(1) = |S|$

$$\Phi_S(1) = \sum_{\sigma \in S} = 1^{w(\sigma)} = |S|$$

$\Phi'_S(1) =$ **sum of weights**

$$\Phi'_S(1) = \sum_{\sigma \in S} = w(\sigma) 1^{w(\sigma)-1}$$

$\Phi'_S(1)/\Phi_S(1)$ **average weight** Follows from above.

1.8 Sum and Product Lemma (Generating Series)

Sum Lemma Suppose $S = A \cup B$, A and B are disjoint. Then

$$\Phi_S(x) = \Phi_A(x) + \Phi_B(x)$$

Product Lemma Suppose $S = A \times B$. Then

$$\Phi_S(x) = \Phi_A(x) \cdot \Phi_B(x)$$

1.9 11 Steps to Generating Series Problems

To solve a generating series problem, we follow 11 steps:

Example. How many compositions of n of k parts?

1. Do you even need generating series?

   Example. Yes.

2. Identify parameters in problem and constants to be treated as parameters.

   Example. n and k

3. Define the set of configurations S by removing one of the parameters.

   Example. Remove n, focus on k parts. Let S be compositions of k parts.

4. Define S in terms of simpler unions and Cartesian products.

Example. $S = (\mathbb{N}_{\geq 1})^k$, where $\mathbb{N}_{\geq 1} = \{1, 2, 3, \ldots\}$

5. Reintroduce removed parameter as weight function $w(\sigma)$ on S.

   Example. n: compositions that have weight n. Thus for $\sigma = (\alpha_1, \ldots, \alpha_k)$ we define $w(\sigma) = \alpha_1 + \ldots + \alpha_k$ or the sum of its parts.

6. Define weight function on simpler sets.

   Example. $\alpha \in \mathbb{N}_{\geq 1}$ is a positive integer. Thus $w(\alpha) = \alpha$.

7. Check weight functions behave correctly for product lemma.

   Example. Note for any $\sigma = (\alpha_1, \ldots, \alpha_k)$, the following holds

   $$w(\sigma) = w_{\mathbb{N}_{\geq 1}}(\alpha_1) + \ldots + w_{\mathbb{N}_{\geq 1}}(\alpha_k)$$

8. Compute generating series of simpler sets.

   Example.

   $$\Phi_{\mathbb{N}_{\geq 1}}(x) = \sum_{\alpha \in \mathbb{N}_{\geq 1}} x^{w_{\mathbb{N}_{\geq 1}}(\alpha)} = \sum_{\alpha \in \mathbb{N}_{\geq 1}} x^{\alpha} = x + x^2 + x^3 + \ldots$$

9. Use Sum and Product Lemma to get formula for $\Phi_S(x)$.

   Example.

   $$\Phi_S(x) = \sum_{\sigma \in S} (\Phi_{\mathbb{N}_{\geq 1}}(x))^k = (x + x^2 + x^3 + \ldots)^k$$

10. Simplify $\Phi_S(x)$.

    Example.

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

    by the infinite geometric series.

11. Answer is the coefficient of x^n or $[x^n]\Phi_S(x)$ where n is your removed parameter.

    Example.

    $$[x^n]\Phi_S(x) = [x^n]x^k(1-x)^{-k} = [x^{n-k}](1-x)^{-k}$$

    Using the Negative Binomial series, $(1-x)^{-k} = \sum_{i=1} \binom{k+i-1}{k-1} x^i$. Note $i = n - k$ thus the coefficient is $\binom{k+(n-k)-1}{k-1} = \binom{n-1}{k-1}$ ways.

1.10 Formal Power Series (FPS)

We can **add/subtract** and **multiply** power series $A(x) = a_0 + a_1x + \dots$ and $B(x) = b_0 + b_1x + \dots$ for rational numbers a_i, b_i.

Addition

$$A(x) + B(x) = \sum_{i\geq 0}(a_i + b_i)x^i$$

Multiplication

$$A(x)B(x) = \sum_{j\geq 0}(\sum_{i=0}^{j} a_ib_{j-i})x^j$$

Note both FPS $A(x), B(x)$ must start at the same index. One can use variable substitution to align them.

2 Counting Binary Strings

A binary string is a string consisting of only **0's and 1's**. In general, we are interested in all binary strings that match a given description of length n.

We will eventually use generating series to count binary strings.

2.1 Concatenations

For two given sets of binary strings A, B, we define the concatenation as

$$AB = \{ab : a \in A, b \in B\}$$

Example. So for $A = \{10, 0\}, B = \{01, 1\}$

$$AB = \{1001, 101, 001, 01\}$$

There is also the * operator on a set of binary strings A where

$$\begin{aligned} A^* &= \{\epsilon\} \cup A \cup AA \cup AAA \cup \dots \\ &= \{\epsilon\} \cup A \cup A^2 \cup A^3 \cup \dots \end{aligned}$$

or it is the concatenation of any number of strings in A. ϵ is the empty string.

Note $\{0, 1\}^*$ is the set of all binary strings.

2.2 Unambiguous Expressions

With concantenation of sets AB we may end up with duplicate elements from two different configurations.

Example. With $A = \{0, 01\}, B = \{11, 1\}$, note

$$AB = \{011, 01, 0111\}$$

where 011 could come from either $0 \in A, 11 \in B$ or $01 \in A, 1 \in B$.

The above expression is **ambiguous**. Formally, an expression is **unambiguous** *if and only if*

$$|AB| = |A \times B|$$

In general, we prefer unambiguous expressions since we can apply the Sum and Product Lemmas to them for generating series.

2.3 0 and 1-Decompositions

For 0-decompositions, we express our set of binary strings S by fixing our 0 characters in the string. Similarly we can do the same for 1 characters with 1-decompositions.

Example. To express all binary strings, we imagine fixing every 0 character ($\{0\}$). 0 or more 1 characters can occur before each 0 ($\{1\}^*$). There may be 0 or more of these 0 preceded by 1s blocks ($(\{1\}^*\{0\})^*$). Since we delimited the expression with the 0 always at the end of string, we need to account for binary strings that end with 1s ($\{1\}^*$). Thus we get the final exprssion

$$(\{1\}^*\{0\})^*\{1\}^*$$

This is called the 0-decomposition. Similarly for 1-decomposition

$$(\{0\}^*\{1\})^*\{0\}^*$$

These two expressions are unambiguous by construction.

2.4 Block Decompositions

A **block** is a maximal substring of 0s or 1s. That is, they are the longest contiguous sections composed of the same character of the binary strings. For the string 000110001000, we separate it out into its blocks

$$000|11|000|1|000$$

Similar to 0 and 1-decompositions, we can decompose binary strings by fixing the blocks of 0s (and similarly 1s). A block of 0s can be expressed as $\{0\}\{0\}^*$ (or $\{0\}^*\{0\}$).

Example. To express the set of all binary strings, we fix every block of 0s. If we follow the same procedure as we did for 0-decompositions, every block of 0s can be preceded by 1s. Note however that the following is ambiguous

$$(\{1\}^*\{0\}\{0\}^*)^*$$

since we need to deliminate the blocks of 0s with a non-empty character (inside the parentheses). For example, 000 is ambiguously constructed in more than 1 way: $(\{\epsilon\}\{0\})(\{\epsilon\}\{00\})$ or $(\{\epsilon\}\{00\})(\{\epsilon\}\{0\})$.

Instead of just 0 or more 1s, we associate each block of 0s by preceding them with a block of 1s. Thus we get the final unambiguous expression for block decomposition

$$\{0\}^*(\{1\}\{1\}^*\{0\}\{0\}^*)^*\{1\}^*$$

Note that we added an extra $\{0\}^*$ and $\{1\}^*$ before and after the expression, respectively. This is necessary since we mandated that the expression in the parenthese begins with a 1-block and end with a 0-block, whereas binary strings can start and end with 0s and 1s, respectively.

Note we can swap the 1s and 0s to get another expression for block decomposition.

2.5 Subsets of Decompositions

Note expressions that are either subsets of the 0 or 1-decompositions or the block decompositions are unambiguous. That is

$$(\{1, 11, 111\}\{0\})^*\{1\}^*$$

is unambiguous since it is a subset of the 0-decomposition where $\{1, 11, 111\} \subset \{1\}^*$.

2.6 Generating Series for Binary Strings

In general, the weight function $w(\sigma)$ of a binary string σ can defined as its length. For example, $w(010) = 3$.

Given that a binary string expression for a set of binary strings S is **unambiguous**, we can apply the Sum and Product Lemmas to find the generating series $\Phi_S(x)$:

Sum Lemma For $S = A \cup B$ (e.g. $\{1, 11\}$ where $A = \{1\}$ and $B = \{11\}$), A and B are disjoint, we have

$$\Phi_S(x) = \Phi_A(x) + \Phi_B(x)$$

Product Lemma For $S = AB$ (i.e concatenation) and given the expression is unambiguous, we have

$$\Phi_S(x) = \Phi_A(x)\Phi_B(x)$$

"Power" Lemma For $S = A^*$, we have

$$\Phi_S(x) = (1 - \Phi_A(x))^{-1}$$

a direct result of the Sum Lemma and the infinite geometric series.

Example. Express and simplify the generating series for the block decomposition expression $S = \{1\}^*(\{0\}\{0\}^*\{1\}\{1\}^*)^*\{0\}^*$.

Note the expression is unambiguous (it is the block decomposition of all binary strings). Use the Sum, Product, and "Power" Lemmas:

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

For a quick sanity check, by the negative binomial theorem

$$\begin{aligned}
(1-2x)^{-1} &= \sum_{i \geq 0} \binom{1+i-1}{i} (2x)^i \\
&= \sum_{i \geq 0} 2^i x^i \\
\therefore [x^n](1-2x)^{-1} &= 2^n
\end{aligned}$$

which aligns with our intuition about the number of binary strings of length n.

3 Coefficients of Recurrence Relations

3.1 Coefficients of Rational Functions

Previously we expressed the generating series as a rational function. In order to find the coefficient $[x^n]$, we can use partial fractions.

In general, for a rational function $\frac{f(x)}{g(x)}$ where $deg(f) < deg(g)$ and

$$g(x) = \Pi_i (1 - \theta_i x)^{m_i}$$

that is we factor $g(x)$ into its linear roots with multiplicities m_i, we have

$$[x^n]\frac{f(x)}{g(x)} = \sum_i P_i(n)\theta_i^n$$

such that $deg(P_i) < m_i$ (that is $P_i(n)$ is a polynomial with degree 1 or more less than m_i).

Example. For the rational function $\frac{1}{(1-2x)(1-3x)}$, we have

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

by the partial fraction decomposition and the negative binomial theorem.

3.2 Coefficients of Recurrence Relations

Let $C(x) = \sum_{n\geq 0} c_n x^n$ where the coefficients c_n satisfy the recurrence relation

$$c_n + q_1 c_{n-1} + \ldots + q_k c_{n-k} = 0$$

We define the characteristic polynomial $g(x)$ as:

$$g(x) = x^k + q_1 x^{k-1} + \ldots + q_{k-1}x + q_k$$

with roots β_i and multiplicity m_i for $i = 1, \ldots, j$ (j roots).

For this recurrence we have the general solution for c_n (the coefficient at x^n, that is $[x^n]C(x)$)

$$c_n = P_1(n)\beta_1^n + \ldots + P_j(n)\beta_j^n$$

where $deg(P_i) < m_i$ and β_i are the roots of the characteristic polynomial $g(x)$.

We can solve for the coefficients of $P_i(n)$ by using the initial conditions and the remainder factor theorem on our general solution.

Example. Find c_n explicitly where

$$c_n - 4c_{n-1} + 5c_{n-2} - 2c_{n-3} = 0$$

for $n \geq 3$ with initial conditions $c_0 = 1, c_1 = 1, c_2 = 2$.

Note the characteristic polynomial for this recurrence is

$$g(x) = x^3 - 4x^2 + 5x - 2 = (x-1)^2(x-2)$$

Therefore the general solution is

$$c_n = (A + Bn)1^n + C \cdot 2^n$$

By the remainder factor theorem

$$\begin{aligned}
c_0 = 1 &= (A + B(0))1^0 + C \cdot 2^0 = A + C \\
c_1 = 1 &= (A + B(1))1^1 + C \cdot 2^1 = A + B + 2C \\
c_2 = 2 &= (A + B(2))1^2 + C \cdot 2^2 = A + 2B + 4C
\end{aligned}$$

Solving the system of equations we get $A = 0, B = -1, C = 1$. Thus we have

$$c_n = -n1^n + 2^n = 2^n - n$$

for $n \geq 0$.