MATH 239: Introduction to Combinatorics

Spring 2022

Sajid Bashar

ENUMERATION

1 Counting

1.1 Definitions and Techniques

$\cup$ is OR, $\cap$ is AND. We can consider expressions $\overbrace{(A \cup B)}$ and $\overbrace{(A \cap B)}$ each as sets themselves.

$|A \cap B| = |A| \times |B|$, $|A \cup B| = \{x : x \in A \text{ or } x \in B\} = |A| + |B| - |A \cap B|$ (account duplicates).

For $\{1, \ldots, n\} = [n]$, there are $n!$ permutations, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ k-sized subsets, and 2^n subsets.

Example. Binary Strings with Exactly k 1s.

For n positions $s_1, \ldots, s_n$, we can select k positions to have 1s. We can then fill the leftover positions with 0s, which does not cost anything since we know for certain where to place them, giving us no choices but to fill the empty positions. So there are $\binom{n}{k}$ of these binary strings.

Some helpful binomial coefficient identities:

- $\binom{n}{k} = \binom{n}{n-k}$ since choosing k is equivalent to not choosing k.
- $\binom{n}{0} = 1 = \binom{n}{n}$
- $\binom{n}{2} = \frac{n(n-1)}{2}$
- $n \leq \binom{n}{2}$ since $\binom{n}{2} \sim \Theta(n^2)$
- $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$

For a tuple λ with length k, $\lambda = (p_1, \ldots, p_k) \in \mathbb{N}^k$.

For n elements and t types, there are $\binom{n+t-1}{t-1}$ ways to form n.

We can create a counting story with a meaning that makes sense to us, and encode this combinatorially.

Sets $S_0, \ldots, S_n$ partition S if $S = S_0 \cup \cdots \cup S_n$ and $S_0 \cap \cdots \cap S_n = \emptyset$. So $|S| = \sum_{k=0}^{n} |S_k|$

1.1.1 Cartesian Product

Pairing of all elements in S and T together into a 2-sized tuple. For all $a \in S$ and $b \in T$, we have

$$\{(a, b) : a \in S \text{ and } b \in T\}$$

The cartesian product is not commutative! So the order of the sets matter s.t. $A \times B \neq B \times A$.

1.1.2 S^* (Infinite Unions)

$$S^* = S^0 \cup S^1 \cup S^2 \cup \cdots = \bigcup_{k \geq 0} S^k$$

where $S^0 = \{\epsilon\}$ and S^k is taking the cartesian product k times s.t. $\lambda = (p_1, \ldots, p_k) \in S^k$.

1.2 Combinatorial Proofs

1.2.1 Combinatorial Interpretation (Double Counting)

The LHS and RHS are interpreted as counting some mutual set S. We simply just express what each side does with S. Since they both count S in their respective ways, they must be equal.

Example. Give a combinatorial proof that $\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{r-k}\binom{n}{k}$.

Proof. Let A and B be disjoint sets s.t. $|A| = m$, $|B| = n$. Then $\binom{m+n}{r}$ is number of r-element subsets of $(A \cup B)$. We can also create an r-element subset of $(A \cup B)$ in the following ways:

- Choose r elements from A and 0 elements from B: We have $\binom{m}{r}\binom{n}{0}$ many subsets.

OR
- Choose $r-1$ elements from A and 1 elements from B: We have $\binom{m}{r-1}\binom{n}{1}$ many subsets.

OR
- $\vdots$

OR
- Choose 0 elements from A and r elements from B: We have $\binom{m}{0}\binom{n}{r}$ many subsets.

Therefore $\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{r-k}\binom{n}{k}$ since we have counted $A \cup B$ in two ways. □

1.2.2 Bijections

These are useful for giving proofs where the LHS and RHS count two different sets, S and S'.

For sets A and B, we can say that the LHS and RHS of an expression are equal if we can map each element of A to B and vice-versa (mutual bijection). Formally, $A \leftrightharpoons B$ if $f : A \to B$ and $g : B \to A$.

We need to show that it's well-defined: $\forall(a \in \mathcal{A}, b \in \mathcal{B})$, then $g(f(a)) = a$ and $f(g(b)) = b$.

Binary strings and ternary strings are an ideal set to biject to since they make for good counting.

Tip: Removing an element that exists and adding it in if it doesn't exist or some variant is common. Also, use venn diagrams for sets to visualize them and try small examples ($n = 3$ or $n = 4$).

Example. Give a combinatorial proof that $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.

Let $S = \{1, \ldots, n\}$ and $S' = \{1, \ldots, n-1\}$. Let A contain all k-element subsets of S and B contain all $(k-1)$-element subsets or k-element subsets of S'. $|A| = \binom{n}{k}$ and $|B| = \binom{n-1}{k-1} + \binom{n-1}{k}$ by definition, so we prove the claim by showing that $A \leftrightharpoons B$.

Define $f : A \to B$ as,

$$f(X) = \begin{cases} X & \text{if } n \notin X \\ X \backslash \{n\} & \text{if } n \in X \end{cases}$$

Define $g : B \to A$ as,

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

By the construction of these functions, $g(f(X)) = X$ for all $X \in A$ and $f(g(Y)) = Y$ for all $Y \in B$. Therefore f is a bijection, so $A \leftrightharpoons B$ and thus $|A| = |B|$.

2 Generating Series (Generating Functions)

2.1 Formal Power Series

2.1.1 Basic Properties

Represent sequence $\{a_n\}$ with $A(x) = \sum_{k\geq 0} a_k x^k$ s.t. a_k are coefficients and x^k are indeterminates.

Two FPS are same iff they have same coefficients for same powers.

$[x^n]$ is the *coefficient of* operator. The extraction itself is by a conversion to a FPS. Some properties:

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

FPS are always invertible. The inverse $A^{-1}(x) = B(x)$ iff $A(x)B(x) = 1$ and exists iff $[x^0]A(x) \neq 0$.

2.1.2 Coefficient Extraction

Example. $[x^{23}]\frac{1+x^2}{(1-2x^3)^7}$

$$[x^{23}](1+x^2)\frac{1}{(1-2x^3)^7} = [x^{23}]\frac{1}{(1-2x^3)^7} + [x^{23}]x^2\frac{1}{(1-2x^3)^7} = [x^{23}]\frac{1}{(1-2x^3)^7} + [x^{21}]\frac{1}{(1-2x^3)^7}$$

By N.B.T.,

$$[x^{23}]\sum_{n\geq 0}\binom{n+6}{6}(2x^3)^n + [x^{21}]\sum_{n\geq 0}\binom{n+6}{6}(2x^3)^n$$

$3 \not| 23$, but $3 \mid 21$ so we can extract $[x^{21}]$ from the second sum, yielding $2^7\binom{13}{6}$.

For a more general case, we can write $[x^n]$ as a piecewise.

Example. $[x^n]\left(\frac{x^2}{1-x^3}\right)^{2022}$

$$[x^n]x^{4044}\frac{1}{(1-x^3)^{2022}} = [x^n]\sum_{n\geq 0}\binom{n+2021}{2021}x^{3n+4044} = \begin{cases}\binom{k+2021}{2021} & \text{if } n = 3k+4044, k \in \mathbb{N} \\ 0 & \text{otherwise}\end{cases}$$

Example. $[x^n]\frac{x^{k-1}}{(1-x)^k}$

$$[x^{n-(k-1)}]\frac{1}{(1-x)^k} = \binom{(n-k+1)-k+1}{k-1} = \binom{n}{k-1}$$

2.2 Defining a Generating Series for S

For an arbitrary set S, we use a weight function w to map each element of S to some meaning. The result is a generating series that "generates" information about S. More precisely,

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

The weight function of A cannot have infinite weight n. Also, cannot have $w(\sigma) = 0$ for A^*.

The coefficient $[x^n]$ tells us the number of elements of S that exactly have weight $w(\sigma) = n$.

The generating series can be finite S_n or infinite S. We can use these series to help model these:

- $(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$ Binomial Theorem
- $\frac{1}{(1-x)^k} = \sum_{n \geq 0} \binom{n+k-1}{k-1} x^n$ Negative Binomial Theorem
- $\frac{1}{1-x} = \sum_{n \geq 0} x^n = 1 + x + x^2 + \ldots$ Geometric Series
- $\frac{A}{1-R} = (AR^0) + (AR^1) + (AR^2) + \ldots$ Generalized Geometric Series
- $\frac{1-x^{n+1}}{1-x} = \sum_{k \geq 0} x^k - \sum_{k \geq n} x^k = 1 + x + \cdots + x^n$ (Finite Geometric Series)
- $(1+x)^\alpha = \sum_{n \geq 0} \binom{\alpha}{n} x^n$ Binomial Series

Tip: $\frac{1}{1+x} = \frac{1}{1-(-x)} = \sum_{n \geq 0} (-x)^n$

Shifting $\sum_{n=0}^{m}$: Left by $\alpha \rightsquigarrow x^{n+\alpha}$, Right by $\alpha \rightsquigarrow x^{n-\alpha}$. Remember to apply this $n \pm \alpha$ everywhere!

2.3 Sum, Product, and String Lemmas

Sum Lemma gives us the generating series of the disjoint union of two sets S and T, more precisely

$$\Phi_{S \cup T}(x) = \Phi_S(x) + \Phi_T(x)$$

Product Lemma gives us the generating series of the cartesian product of S and T, more precisely

$$\Phi_{S \times T}(x) = \sum_{a \in S} \sum_{b \in T} x^{w_S(a) + w_T(b)} = \sum_{a \in S} x^{w_S(a)} \cdot \sum_{b \in T} x^{w_T(b)} = \Phi_S(x) \Phi_T(x)$$

String Lemma only holds if there are no elements in S with weight 0. That is,

$$\Phi_{S^*}(x) = \frac{1}{1 - \Phi_S(x)}$$

2.4 Integer Compositions

2.4.1 Properties

Composition of n is an ordered sequence $\overbrace{(c_1, c_2, \ldots, c_k)}$ of k positive integers s.t. $c_1 + \cdots + c_k = |C| = n$.

We say S contains these compositions $\sigma = (c_1, \ldots, c_k)$ (usually with constraints) of all possible n, and use the Product Lemma (above for visualization) along with the weight function:

$$w(\sigma) = \sigma \implies w((c_1, \ldots, c_k)) = w_{c_1}(c_1) + \cdots + w_{c_k}(c_k) = c_1 + \cdots + c_k \implies w(C) = |C|$$

to find an explicit formula, which works because of the Product Lemma. Remember that 0 cannot be a weight! So say if a part is even, then $E = \{2, 4, \dots\}$. Then extracting $[x^n]$ will give us how many compositions exist for a specific n since we're generating a number set.

We can consider any S as k blocks, S_k, s.t. blocks are generated independently, then package them together using $\cap$ and $\cup$, and then use the Product Lemma on all k blocks for $\Phi_{S_k}(x)$.

To generate all possible k, we use the String Lemma on $\Phi_{S_k}(x)$ since we need to generate all sizes.

We're essentially deconstructing each k-tuple (composition) and deducing how they were generated.

2.4.2 Algorithm for Finding Integer Compositions

(1) Determine respective generating series that satisfy constraints of each part.

(2) Package all generating series found in (1) into one generating series for k parts given by $\Phi_{S_k}(x)$. Consider any constraints on k s.t. k implicitly implies a different k. Can there be 0 parts?

> **Example.** Empty Compositions
>
> "The first part is odd" implies that the first part exists so $k \implies k+1$. "All parts are odd" does not imply that a part exists. "There is an even number of parts" does not imply that a part exists. "The last part is divisible by 3" implies that the last part exists.

(3) If unknown number of parts, use String Lemma to generate all possible k. So $\Phi_S(x) = \sum_{k \geq 0} \Phi_{S_k}(x)$.

(4) If needed, extract $[x^n]\Phi_S(x)$ to find all possible integer compositions of n.

Tip: Of k parts, if exactly 2 parts are even $\implies k-2$ parts are odd. Instead, if at least 2 parts are even $\implies k-2$ parts are $\mathbb{N}$. Same idea works if we swap which parts are even and odd.

Tip: Congruence relation is given by $a \equiv b \pmod{n} \implies a = bk + n$.

3 Binary Strings

Blocks are maximal substrings of one digit, like $\underbrace{0000}\underbrace{111}$.

3.0.1 Concatenation

Concatenation of two strings A and B is analogous to taking the cartesian product but removing commas. More formally, $\{ab : a \in A \text{ and } b \in B\}$, like FOIL. Careful, this isn't commutative so it may cause ambiguity!

Example. $A = \{0, 01, 011\}$, $B = \{1, 11\}$

$$AB = \{01, 011, 011, 0111, 0111, 01111\} \implies \text{Ambiguous}$$
$$BA = \{10, 101, 1011, 110, 1101, 11011\} \implies \text{Unambiguous}$$

3.1 Regular Expressions

Each regular expression R recursively produces a corresponding set $\mathcal{R}$ of binary strings (rational language). Regex consists of,

- $\epsilon, 0$, or 1
- $R_1 \smile R_2$ (Union) (Choose R_1 or R_2)
- R_1R_2 (Concatenation) (R_1 followed by R_2)
- R^* (Concatenation Powers) (Include R arbitrary times)

Informally, we can interpret regex as a process to build a specific string in $\mathcal{R}$.

Tip: Treat $(R_1 \smile R_2)^*$ as infinite iterations, in each iteration we arbitrarily choose which to concatenate.

Example. $\mathcal{S}$ contains all strings s.t. each block of 1's is followed by a block of 0's of length one or four. Write an unambiguous regular expression that produces $\mathcal{S}$.

$$0^*(11^*00^*)^*1^* \implies 0^*(11^*(0 \smile 0000))^*$$

We can also define a recursive regex in terms of itself:

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

Can also write $\{\epsilon\} \cup S\{0, 1\}$. This produces all binary strings, and reads as "A string produced by S is the empty string or a string produced by S followed by 0 or 1".

3.2 Ambiguity

Regular expression R is unambiguous if every string is produced exactly once (uniquely formed) in $\mathcal{R}$. Otherwise, R is ambiguous.

3.2.1 Proving Ambiguity

Give example of a specific string that is produced in more than one way.

Example. Show that $S = (11)^*10 \smile (11)^*11 \smile 0)^*(\epsilon \smile 1)$ is ambiguous.

We can produce the binary string $\alpha = 1111111111$ in two ways using the expression $((11)^*10 \smile (11)^*11 \smile 0)\epsilon$. Note that we choose to concatenate the empty string using the expression $(\epsilon \smile 1)$. The first way to produce α is by first choosing $\epsilon(11)$ from the nested expression $(11)^*11$, and then concatenating $(111111)11$ also from $(11)^*11$. That is,

$$\underbrace{11}\underbrace{11111111}$$

The second way to produce α is by first choosing $(11)11$ from the nested expression $(11)^*11$, and then concatenating $(1111)11$ also from $(11)^*11$. That is,

$$\underbrace{1111}\underbrace{111111}$$

3.2.2 Proving Unambiguity

Show REs separated by $\smile$ are disjoint, then we can say that RE is $\therefore$ unambiguous.

Show concatenated REs $\leftrightharpoons$ with Cartesian product of rational languages they produce (*Lem* 3.9).

Example. Show that $S = (001 \smile 01 \smile 1)^*0^*$ is unambiguous.

Each string produced has the form $\alpha\beta$ where $\alpha = (001 \smile 01 \smile 1)^*$ and $\beta = 0^*$. Notice that 001, 01, and 1 are disjoint and thus $01 \smile 001 \smile 1$ is unambiguous. α does not produce 0, so β is unambiguously all copies of 0 at the end of the string. All strings formed by α are of the form

$$\underbrace{11...1}_{k}\mu\underbrace{11...1}_{k_1}\mu\underbrace{11...1}_{k_2}...$$

where $\mu = 0$ or $\mu = 00$, $k \geq 0$, and $k_i \geq 1$. Given k, $k_1, k_2, \ldots$ there is exactly one way to produce the string above that form from $(001 \smile 01 \smile 1)^*$. All strings produced by 0^* have the form $000\ldots$ where 0 is repeated $k' \geq 0$ times and so there is exactly one way to produce the string from 0^*.

3.3 Block Decomposition

We break down binary strings by decomposing them into blocks. To help us make unambiguous expressions, we can alter a known unambiguous expression that produces all binary strings in blocks by treating it like a template and substituting for each of the parts depending on $\mathcal{S}$ (*Cor.*):

$$0^*(11^*00^*)^*1^* \text{ or } 1^*(00^*11^*)^*0^*$$

Other unambiguous expression templates include: $(0 \smile 1)^*$, $0^*(10^*)^*, 1^*(01^*)^*$

Tip: If constraint, for first and last block give option of using ϵ.

3.4 Recursive Block Decomposition

3.4.1 The Forbidden Substring α

Proof Sketch. Let S be the set of all binary strings that do not contain α as a substring, and let T be the set of strings that have exactly one occurrence of α, at the very end (that is, as a suffix). A string in $S \cup T$ is either empty, or it ends with either a 0 or a 1. Translating this into a relation is given as

$$S \cup T = \{\epsilon\} \cup S\{0,1\}$$

which corresponds to the generating series of

$$\Phi_S(x) + \Phi_T(x) = 1 + \Phi_S(x)(2x) \tag{1}$$

Notice that the string α (= 1011101 for e.g.) can overlap itself in a nontrivial way. That is,

.	.	.	.	.	.	1	0	1	1	1	0	1
1	0	1	1	1	0	1	↑	↑	↑	↑	↑	↑
.	1	0	1	1	1	$\underline{0}$	$\underline{1}$	.	.	.	.	.
.	.	1	0	1	1	1	0	1	↑	↑	↑	↑
.	.	.	1	0	1	$\underline{1}$	$\underline{1}$	$\underline{0}$	$\underline{1}$	.	.	.
.	.	.	.	1	0	$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{0}$	$\underline{1}$	.	.
.	.	.	.	.	1	$\underline{0}$	$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{0}$	$\underline{1}$	.

Looking at all the possible ways that α can overlap itself, we see that in this case

$$S\{\alpha\} = T\{\epsilon, \beta_1, \ldots, \beta_k\} \tag{2}$$

By definition $T\{\epsilon, \beta_1, \ldots, \beta_k\} \subseteq S\{\alpha\}$. To verify $S\{\alpha\} \subseteq T\{\epsilon, \beta_1, \ldots, \beta_k\}$, we have to consider if S can end with a proper prefix of α (not all) s.t. when S is concatenated with α we may get two αs (overlapping). All the bits on the LHS of the table are the possible endings that strings in S can have. If all bits on the RHS match up with α (i.e. we can make two copies of α), then take the bits with arrows and concatenate them and add them to the list of βs so T also produces two αs.

3.5 Parsing Regular Expressions into Generating Series

For a length function ℓ, always define $w(\sigma) = \ell(\sigma)$. So $[x^n]\Phi_{\mathcal{S}(x)}$ is the number of binary strings of rational language $\mathcal{S}$ (made by S) with length n. Factoring $\Phi_{\mathcal{S}}(x)$ is helpful for parsing recursive regex.

If $R = f(x)$ and $S = g(x)$, the parsing rules are:

- $\epsilon \rightsquigarrow x^0 = 1$
- $0,\ 1 \rightsquigarrow x^1$
- $R \smile S \rightsquigarrow f(x) + g(x)$
- $RS \rightsquigarrow f(x)g(x)$
- $R^* \rightsquigarrow \frac{1}{1-f(x)}$

Example. Find $\Phi_{\mathcal{S}}(x)$ of $S = 0^*(11^*(0 \smile 0000))^*$.

$$\Phi_{\mathcal{S}}(x) = \frac{1}{1-x} \cdot \frac{1}{1 - (x \cdot \frac{1}{1-x} \cdot (x + x^4))}$$

4 Recurrence Relations

We use linear recurrences $\{a_n\}$ to help us find coefficients of rational functions $\frac{P(x)}{Q(x)}$. These sequences satisfy $\{a_n\} = 0$.

In particular, when written in terms of a_n, then $a_n \iff [x^n]A(x)$.

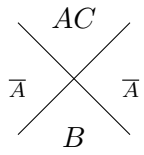
Remember that the recurrence should be in terms of just a_n (including not being $-a_n$). When dealing with any computation, always re-index the recurrence so that it relies on previous terms and not terms ahead.

Example.

$$a_{n+2} = 2a_n + a_{n+1} \implies a_n = 2a_{n-2} + a_{n-1}$$

4.0.1 Factoring Polynomials of Degree 2

Always factor out (-1) from $-a_2x^2$.



4.0.2 Factoring Polynomials of Degree 3

Say $a_3x^3 \pm a_2x^2 \pm a_1x \pm a_0$, try finding a root $x = \alpha$, then use synthetic division.

$$\begin{array}{r|rrrr} & -2 & -3 & 0 & 1 \\ -1 & & 2 & 1 & -1 \\ \hline & -2 & -1 & 1 & 0 \end{array}$$

4.0.3 Partial Fractions

(1) Factor $Q(x)$. Assign a partial fraction for each exponent from 1 up.

(2) Try substituting roots and solving for unknown constants. Otherwise, solve system of equations.

4.1 Rational Power Series to Explicit Coefficient Formula $[x^n]$

(1) Factor $Q(x)$.

(2) Partial fraction decomposition to create linearly independent RPS.

(3) Simplify, remember to preserve $[x^n]$ throughout calculation.

Example. Find an explicit formula of $A(x) = \frac{3-20x}{1-10x+25x^2}$.

$$\begin{aligned}
[x^n]\frac{-4}{5x-1} + \frac{-1}{(5x-1)^2} &= [x^n]\left(4 \cdot \frac{1}{1-5x} - \frac{1}{(1-5x)^2}\right) \\
&= 4[x^n]\sum_{n\geq 0}(5x)^n - 1[x^n]\sum_{n\geq 0}\binom{n+1}{1}(5x)^n \\
&= 4 \cdot 5^n - 1 \cdot (n+1) \cdot 5^n \\
&= 5^n(4-(n+1)) = 5^n(3-n)
\end{aligned}$$

4.2 Rational Power Series to Linear Recurrences

$P(x)$ is initial terms and $Q(x)$ is the equation for the recurrence relation. Using $Thm.\ 4.8$, we read the RHS and give the equation as a piecewise with the convention that $a_n = 0$ if $n < 0$. Parsing algorithm:

(1) Convert $Q(x)$ into a recurrence relation set to zero and match each power of all terms x^k to a_{n-k}.

(2) Correspond each power of $P(x)$ s.t. $\alpha x^k \to$ "α, if $n = k$" in the piecewise ($A(x) = \alpha$). Then, "0, if $n \geq (deg(Q(x)))$" in piecewise.

(3) Find initial conditions up to $a_{deg(Q(x))-1}$ by setting recurrence relation from (2) to each α and solving for each a_k.

Example. $A(x) = \frac{1+x-4x^2}{1-3x^2+2x^3}$

From Theorem 4.8, we can read from the RHS of $A(x)$ that for all $n \in \mathbb{N}$:

$$a_n - 3a_{n-2} + 2a_{n-3} = \begin{cases} 1 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ -4 & \text{if } n = 2 \\ 0 & \text{if } n \geq 3 \end{cases}$$

with the convention that $a_n = 0$ if $n < 0$. We can determine the initial conditions as follows:

$$\begin{aligned}
&[x^0] : a_0 = 1 \\
&[x^1] : a_1 = 1 \\
&[x^1] : a_2 - 3a_0 = -4 \implies a_2 = -1
\end{aligned}$$

The recurrence $a_n - 3a_{n-2} + 2a_{n-3}$ holds for all $n \geq 3$.

4.3 Linear Recurrences to Rational Power Series

(1) Use initial conditions with our convention to find α, then form $P(x)$.

(2) Convert recurrence to $Q(x)$ of rational function by matching a_{n-k} to x^k.

(3) If explicit formula for a_n needed, use 4.1.

Example. $a_n - 10a_{n-1} + 25a_{n-2} = 0$ for $n \geq 2$, $a_0 = 3$, $a_1 = 10$. Find RPS of a_n.

By Theorem 4.8, reading the recurrence yields $Q(x) = 1 - 10x + 25x^2$. With the convention that $a_n = 0$ if $n < 0$ we solve $P(x)$ as,

$$\underline{n = 0} : a_0 = 3$$
$$\underline{n = 1} : a_1 - 10a_0 = 10 - 10(3) = -20$$

So $P(x) = 3 - 20x$, therefore

$$A(x) = \frac{3 - 20x}{1 - 10x + 25x^2}$$

4.4 Linear Recurrences to Formulas of a_n as Functions of n (Explicit $[x^n]$ Formula)

The conversion of the recurrence to $Q(x)$ is identical. We also use $Thm.$ 4.14 which implies that there are constants for sufficiently large n s.t. a_n is a general formula. Then we use given recurrence terms to solve for these constants. The algorithm is:

(1) Convert $Q(x)$ into a recurrence relation set to zero and match each power of all terms x^k to a_{n-k}.

(2) Factor $Q(x)$.

(3) Use (2) to find a general formula a_n with constants s.t. each constant factor is degree $n - 1$ less.

Example. $Q(x) = (1 - 2x)^2(1 + x) \implies a_n = (A + Bn)2^n + C(-1)^n$

(4) Solve for the unknown constants using the initial recurrence terms by setting a_n to each term and solving simultaneous equations.

4.5 Catalan Numbers

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

GRAPH THEORY

5 Properties of Graphs

5.1 Basic Characterizations

$G = (V(G), E(G))$ s.t. V is vertices and E has edges (unordered pairs of vertices).

If u and v are neighbours, they are adjacent ($v \in N_G(u)$) and uv is incident to u and v.

$deg(v)$ is the number of edges incident with v. If $deg(v) = 0$, then v is an isolated vertex.

5.1.1 Matrix Multiplication

Number of rows in A must equal number of columns in B, and their sizes n also need to be the same. c_{ij} is the dot product of the i^{th} row of A and j^{th} column of B where the dot product is multiplying term-by-term and summing them up.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in} + b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

5.1.2 Adjacency Matrix

$$a_{ij} = \begin{cases} 1, \text{if } v_i \text{ is adjacent to } v_j \\ 0, \text{ otherwise} \end{cases}$$

5.1.3 Incidence Matrix

Rows are vertices v_i, columns are edges e_j.

$$b_{ij} = \begin{cases} 1, \text{if } e_j \text{ is incident to } v_i \\ 0, \text{ otherwise} \end{cases}$$

Each row sum corresponds to the degree of a vertex and each column sum is always 2 (edges are incident to two vertices or loops are incident to themselves). The sum of all row sums is twice the number of edges (i.e. number of columns).

5.1.4 Handshaking Lemma

$$2e = \sum_{v \in V(G)} deg(v)$$

This property must hold for all components of G, even if we remove any number of edges.

It also implies there is an even number of vertices with odd degree (*Cor* 4.3.2).

Example. A vertex has degree 1 or 5. Show that $n_1 = \frac{5}{4}n - \frac{1}{2}e$

$$2e = n_1 + 5(n_5) \tag{1}$$

$$n = n_1 + n_5 \implies n_5 = n - n_1 \tag{2}$$

$$2e = n_1 + 5(n - n_1) \implies 2e = n_1 + 5n - 5n_1 \implies 4n_1 = 5n - 2e \implies n_1 = \frac{5}{4}n - \frac{1}{2}e$$

5.1.5 Edge Deletion

We can remove an edge s.t. $G - e$ or $E(G)\backslash\{e\}$.

The Handshaking Lemma should still hold after $G - e$.

5.2 Isomorphisms

Two graphs are isomorphic if one can be obtained from the other by renaming vertices while the structural property is preserved. Formally, it's a bijection $f : V(G) \to V(G')$ where for $v \in V(G)$ we have a corresponding $f(v) \in V(G')$ (for multigraphs, we need to find a bijection for the edges as well).

The isomorphism class is the collection of graphs that are isomorphic to G. Up to isomorphism ($\forall$ classes) depends on the number of configurations of G where the adjacency structures are all different.

5.2.1 Proving Two Graphs are Isomorphic

Directly state a mapping for each vertex $v \in G$ to G' in a table. Note that $f(v)$ is symmetric.

Tip: Compare the degrees and neighbours to identify the mapping.

v	$f(v)$
$\vdots$	$\vdots$

5.2.2 Proving Two Graphs are Not Isomorphic

Identify a graph invariant (property of G that doesn't change under an isomorphism). Helpful ones:

- Degree Sequence (Non-decreasing sequence $(d_1, \ldots, d_n)$)
- Planarity
- Bipartiteness
- Number of Triangles (Copies of K_3 or 3-cycles)
- Number of Cycles with the Same Length

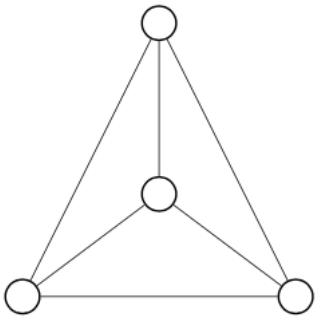
5.3 Classes of Graphs

5.3.1 Complete Graphs - K_n

Each pair of vertices is an edge. All vertices are degree $n - 1$, so there is $\binom{n}{2}$ edges.

A complete bipartite is denoted $K_{m,n}$ where m and n are the respective sizes of vertex classes.

Example. K_4.



5.3.2 K-Regular Graphs

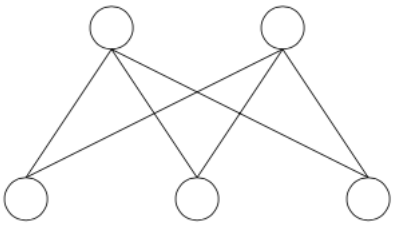
All vertices are degree k, so there are $\frac{nk}{2}$ edges.

5.3.3 Bipartite Graphs

A partition of vertices (A, B) exists where each edge of G joins one vertex in A with B.

$\sum_{v \in A} deg_G(v) = \sum_{v \in B} deg_G(v)$

Example. $K_{2,3}$.



Tip: Counting the number of edges for one vertex class gives us the number of edges for the other.

5.3.4 n-Cube (Hypercube)

$V(G)$ is the set of all binary strings of length n. $N_G(v)$ consists of all binary strings that differ in exactly one position.

Hypercubes are n-regular. All n vertices have degree n, since from n positions we choose $k = 1$ bits to flip which implies that a vertex has $\binom{n}{1} = n$ neighbours. So there's $n \cdot 2^{n-1}$ edges.

As an aside, this can be generalized further by noting that a Hamming distance of k means choosing k bits to be flipped, so that in general, any binary string of length n has $\binom{n}{k}$ other strings at Hamming distance k.

They are also bipartite, A has all vertices with an odd number of 0s and B with all vertices of an even number of 0s.

5.4 Walks and Paths

The length is the number of edges. Trivial walks and paths have length 0.

5.4.1 uv-Walk

Sequence of alternating vertices and edges ($u = v_0, v_0v_1, \ldots, v_{k-1}v_k, v_k = k$).

5.4.2 uv-Path

A uv-walk with no repeated vertices (thus no repeated edges).

If there is a uv-walk in G, there is a uv-path in G. The proof is by finding repeated vertices, removing the parts between the repetition, and repeating until no vertices repeat (use a shorter route).

If an xu-path and uy-path exist, then a xy-path exists (Thm 4.6.2).

5.5 Euler Tours (Euler Circuits)

From a specific v, we visit every edge of G exactly once and return to v (closed walk).

Euler Tours are only possible iff $\forall$ vertices are of even degree (Thm 4.9.2).

5.6 Subgraphs and Cycles

5.6.1 Subgraphs

H is a subgraph if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$.

If $V(H) = V(G)$, H is a spanning subgraph. Also, if $H \neq G$ is a proper subgraph (H is not G itself).

5.6.2 Cycles

A cycle is a subgraph with k distinct vertices. They're 2-regular graphs and are uv-paths + uv.

Cycles must at least have a length of 3 (K_3). If each vertex has at least degree 2, then G contains a cycle. The proof is by the shortest path argument.

If there are two distinct paths from vertex u to vertex v in G, then G contains a cycle ($Cor.$ 4.10.4).

Proof Sketch. Assume $P = v_0, \ldots, v_k$ is the longest path. Since v_0 has degree at least 2, it has another neighbour x. If x is not on the path, then $x, v_0, \ldots, v_k$ is longer, contradiction. So $x = v_i$ for some $i \geq 2$, and $v_0, v_1, \ldots, v_i, v_0$ is a cycle in G. □

The girth is the length of the shortest cycle. If a cycle does not exist, the girth is infinity.

A spanning cycle is a Hamilton Cycle - we have to visit all the vertices exactly once.

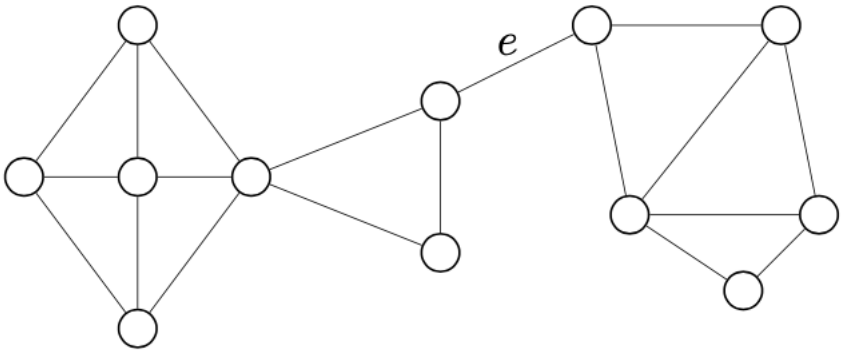
A graph is bipartite iff it has no odd cycles (Thm 5.3.2).

5.7 Connectedness

5.7.1 Components and Bridges

Components are islands without bridges, or more formally maximally connected subgraphs of G.

Bridges connect these islands together, where an edge e is a bridge if $G - e$ gives more components of G (Thm 4.10.3) and is one iff it's not contained in any cycle of G.



The cut induced by $X \subseteq V(G)$ is the set of all edges in G with precisely one end in X. It is essentially all the edges that are bridges between the vertices in whatever the cut is being induced as (i.e. X) with everything else it's connected to. It can be drawn as a line through the edges in the cut.

5.7.2 Connected Graphs

G is connected if for all pairs of vertices u and v, there is a path. So we can get from any point A to B.

If for each vertex v, there is a path from v to u then G is connected. That is, if there's a path between a fixed vertex and each of the remaining vertices we say that G is connected (Thm 4.8.2).

If $V(H_1) \cap V(H_2) \neq \emptyset$, then $H_1 \cup H_2$ is connected s.t. $H_1 \cup H_2 = (V(H_1) \cup V(H_2), E(H_1) \cup E(H_2))$.

Example. Show that the n-cube is connected.

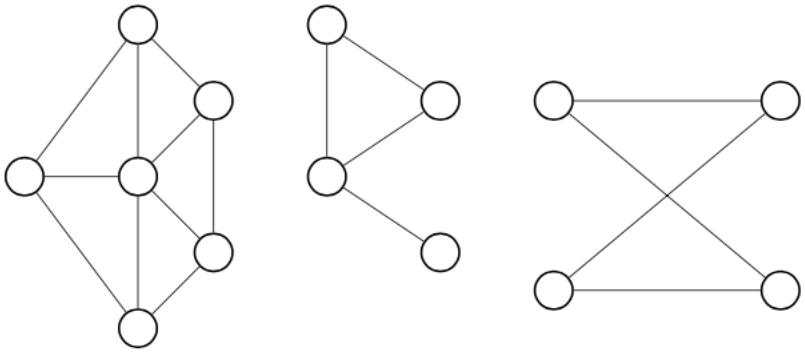
The idea is that we can get to the vertex of n 0's from an arbitrary vertex v. This can be done by continuously flipping the leftmost 1, since $N_{G_n}(v)$ is all binary strings that differ in 1 position.

Proof. Let $v_0 \in V(G)$ be the string of n 0's. Let x be any string of length n. Suppose x has k 1's at positions $i_1, \ldots, i_k$. We produce k strings as follows: v_j is the string of length n with exactly j 1's, at positions $i_1, \ldots i_j$. Then v_j and v_{j+1} differ in exactly one position i_{j+1}. So $v_j v_{j+1}$ is an edge. Then $v_0, v_1, v_2, \ldots, v_k = x$ is a v_0, x-path. By Thm. 4.8.2, the n-cube is connected. □

5.7.3 Disconnected Graphs

G is not connected iff there exists a non-empty proper subset X where the cut induced by X is empty ($\emptyset$) (Thm 4.8.5). So to show G is disconnected, we need to show that the cut induced by X is empty which means there's no edges between X and $V(G) \backslash X$.

This implies that if G is disconnected, there's at least two components.

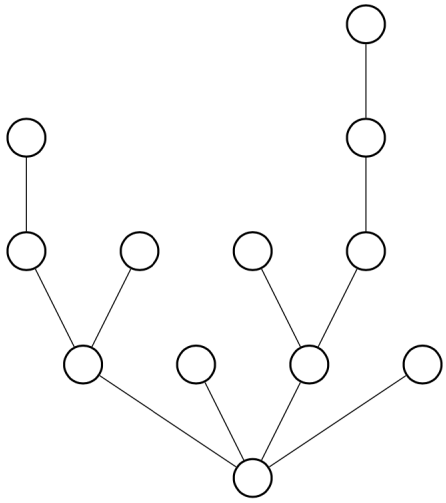


Tip: Imagine isolating each vertex in X using a circle.

6 Trees

6.1 Properties

(Minimally) connected graphs with no cycles.



A forest is multiple trees.

Every edge in T is a bridge ($Lem.$ 5.14). The proof uses the fact that no edge is in a cycle.

$\forall u, v \in V(T)$, there's a unique u, v path in T ($Lem.$ 5.1.3).

For trees, $m = n - 1$ (or $|E(G)| = |V(G)| - 1$) ($Thm.$ 5.15).

If G is connected and has $m = n - 1$ edges, then G is a tree.

More generally, for a forest with k components, then $|E(G)| = |V(G)| - k$ (Cor 5.16).

Leafs are vertices with degree 1.

Every T with at least 2 vertices has at least 2 leaves ($Thm.$ 5.18). The proof is by the longest path argument.

Trees are bipartite. Bipartite graphs cannot contain odd cycles and trees don't have them anyways.

6.2 Spanning Trees

Spanning subgraphs that are trees.

A graph is connected iff it contains a spanning tree (Thm 5.2.1).

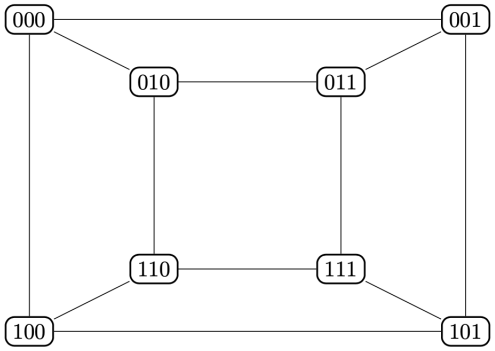
If T is a spanning tree of G and e is an edge not in T, then $T+e$ contains exactly one cycle C. Moreover, if e_0 is any edge on C, then $T + e - e_0$ is also a spanning tree of G (Thm 5.2.3).

If T is a spanning tree of G and e is an edge in T, then $T - e$ has 2 components. If e_0 is in the cut induced by one of the components, then $T - e + e_0$ is also a spanning tree of G (Thm 5.2.4).

7 Planar Graphs

7.1 Properties of Planarity

Graphs where all edges only intersect at their ends. The drawing is a planar embedding of G.



Planar embeddings partition the plane into faces (connected regions). The outer face is unbounded.

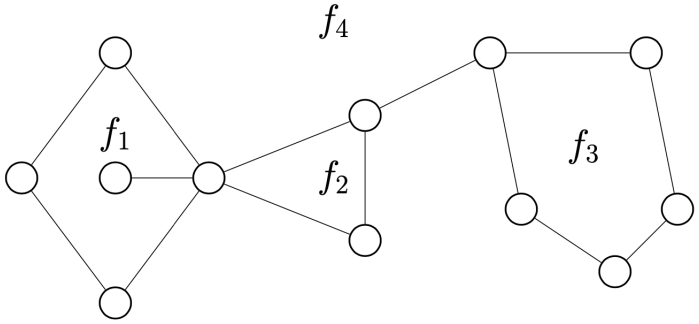
7.1.1 Certifying Planar Graphs

The only way to show that G is a planar graph is to directly provide a planar embedding.

7.1.2 Faces

The boundary of a face is all vertices and edges that touch the face. A boundary walk is a closed walk around the perimeter of the face walk.

The degree of a face is the length of its boundary walk.



Tip: Think of a face as being the result of the bucket function in MS Paint.

7.1.3 Faceshaking Lemma

$$2e = \sum_{f \in F(G)} deg(f)$$

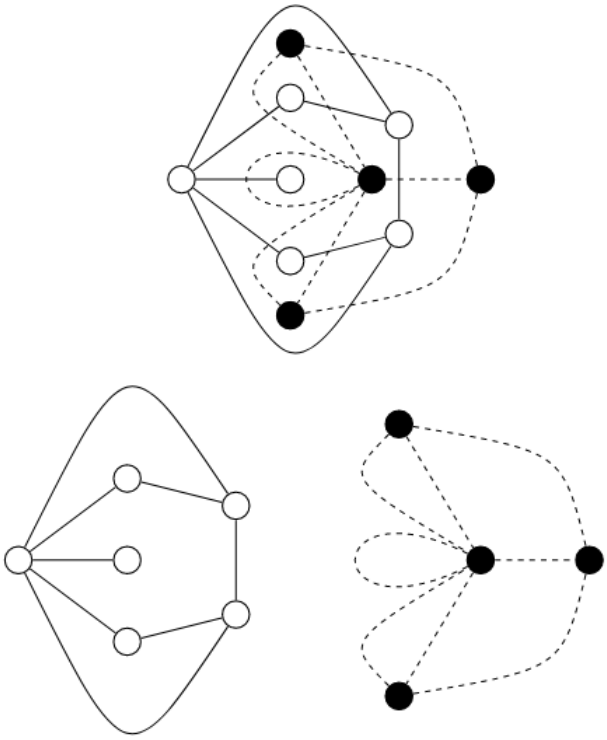
In planar embeddings, e is a bridge iff two sides of e are in the same face.

7.1.4 Jordan Curve Theorem

Every planar embedding separates the plane into two parts, one on the outside (exterior) and one on the inside (interior).

7.2 Duals

A planar embedding has a corresponding dual G^* which is constructed as follows: G^* has one vertex for each face of G. Two vertices of G^* are joined by an edge whenever the corresponding faces of G have an edge in common (one side for each face), and the edge in G^* is drawn to cross this common boundary edge in G.



If G is connected, then $G = (G^*)^*$

7.3 Euler's Formula

7.3.1 Definition

For all connected planar embeddings the following will always hold,

$$n - m + s = 2$$

7.3.2 Euler's Inequality (Formula for Non-connected Graphs)

$$n - m + s = 1 + c \geq 2$$

Proof Sketch. Connecting c components adds $c-1$ edges while not increasing the number of faces. Adding bridges does not change the planarity since e is surrounded by the outer face on both sides. So with our new m, we substitute $n-(m+c-1)+s=2$ which yields the inequality as needed. □

If G is planar s.t. $\forall$ faces have degree at least $d^* \geq 3$, then

$$m \leq \frac{d^*}{d^*-2}(n-2) \qquad (Lem\ 7.5.2)$$

If G is planar and contains a cycle, then the boundary of each face contains a cycle (Lem).

Example. Show that $\forall$ planar embeddings has a vertex of degree ≤ 5 or a face of degree ≤ 3.

Proof. Assume that all faces have degree ≥ 3. Then by Lemma 7.5.2 we have that

$$m \leq \frac{d^*}{d^*-2}(n-2) \implies m \leq \frac{3}{1}(n-2) = 3(n-2)$$

The average degree of a vertex is defined as

$$\frac{\sum_{v\in V(G)} deg(v)}{n} = \frac{2m}{n} \leq \frac{2(3(n-2))}{n} = 6 \cdot \frac{n-2}{n} < 6$$

So the average degree is less than 6, which implies there is a vertex with degree at most 5. □

7.4 Certifying Nonplanar Graphs

7.4.1 Using Inequalities

In a non-bipartite planar G with $p \geq 3$ vertices and q edges, we have

$$q \leq 3p-6$$

In a bipartite planar G with $p \geq 3$ vertices and q edges, we have

$$q \leq 2p-4$$

All planar graphs have a vertex of degree at most 5 (Cor).

So to certify non-planarity, we can show that these inequalities do not hold.

Example. Show that K_5 is a non-planar graph.

Corollary. By the Lemma, $\binom{5}{2} \leq 3(5)-1 \implies 10 \leq 9$

which clearly does not hold. $\therefore$ K_5 is not planar. □

The same proof works for showing that $K_{3,3}$ is not planar as well.

7.4.2 Kuratowski's Theorem

An edge subdivision of G is obtained after each edge is independently replaced by a path of length 1 or more.

Using Kuratowski's Theorem, we can show that a graph is non-planar by showing that G has a subgraph that is an edge subdivision of K_5 or $K_{3,3}$.

Tip: Identifying an odd cycle eliminates the possibility of $K_{3,3}$.

7.5 Colouring and Planar Graphs

A k-colouring of G is an assignment of k colours to all vertices of G where adjacent vertices cannot be assigned the same colour.

A graph with a k-colouring graph is k-colourable. Also, a k-colourable graph is $(k+1)$-colourable, $(k+2)$-colourable, and so on.

7.5.1 Bipartite Graphs

A graph is 2-colourable iff it's bipartite (Thm 7.7.2).

Proof Sketch. Given bipartition (A, B), both directions involves setting all vertices of A to be the first colour and B the second colour. □

7.5.2 6-Colour Theorem

All planar graphs are 6-colourable (Thm).

The proof is dependent on the fact that there exists a vertex with degree at most 5.

Proof. Assume that G is a planar graph. The proof is by induction on n vertices where $P(n)$ is the statement that G is 6-colourable.

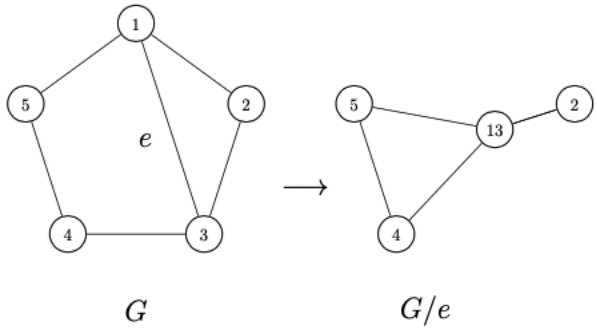
Base Case: All planar graphs on one vertex are 6-colourable, so the result is true for $n = 1$.

Inductive Hypothesis: Assume that $\forall$ planar graphs are 6-colourable for $k \leq n$ vertices, where $k \geq 1$.

Inductive Step: Consider G such that there are $n = k + 1$ vertices. By a Lemma, we know that all planar embeddings have at least one vertex v with $deg(v) \leq 5$. Suppose we remove vertex v, and all edges incident to v from G. We denote this resulting graph as G'. Then G' has k vertices, and is a planar graph since all subgraphs of a planar graph are also planar. We apply the inductive hypothesis to G', so G' is 6-colourable. Now find a 6-colouring of G'. There are at most 5 vertices in G' that are adjacent to v in G, so these vertices are assigned at most 5 different colours in the 6-colouring of G. Thus there is at least one of the 6 colours remaining. Assign one of these remaining colours to v, so that v has a different colour from all of its adjacent vertices in G. Thus we have a 6-colouring of G, and the result is true for $n = k + 1$. We have now proved that the result is true by POMI. □

7.5.3 5-Colour Theorem

Edge contraction for edge $e = x, y$ is an operation that allows the length of e to decrease to 0 s.t. vertices x and y are identified as a new a vertex $z = xy$ and produces a new graph $H = G/e$.



All planar graphs are 5-colourable (Thm).

Proof. Assume that G is a planar graph. The proof is by induction on n vertices.

Base Case: All planar graphs on one vertex are 5-colourable, so the result is true for $n = 1$.

Inductive Hypothesis: Assume that all planar graphs on $n \leq k$ vertices are 5-colourable s.t. $k \geq 1$.

Inductive Step: We consider two cases of G for $n = k + 1$ vertices:

Case I: G has a vertex of degree at most 4

Use same argument as the 6-colour theorem proof.

Case II: G has no vertex of degree at most 4 (G is minimum degree 5)

This implies that G has a vertex v of degree 5 (since $\forall$ planar embeddings have a vertex of degree at most 5 and we assumed that there does not exist a vertex of at most degree 4). Furthermore, all 5 adjacent vertices of v cannot be mutually adjacent as otherwise we would have a K_5 subgraph which contradicts our assumption that G is planar. So there are at least two vertices $x, y \in N_G(v)$ that are not adjacent. Contract edge $\{x, v\}$ to obtain $H = G/e$ and label the new vertex as v, then contract $\{y, v\}$ to obtain $K = H/e$ and label the new vertex as v. As G is planar, we know that H and K is as well; moreover, K has $k+1-2 = k-1$ vertices and thus K is 5-colourable by the inductive hypothesis. Use the 5-colouring of K to colour all the vertices of G except for x, y, and v. We know that x and y are not adjacent in G, so they can be assigned the same colour. Colour both vertices x and y in G with the colour assigned to vertex v in K. Hence, at most four distinct colours appear on the five neighbours of v. Colour v with one of the absent colours. Then we have a valid 5-colouring of G.

In both cases, G has a 5-colouring. Therefore, by mathematical induction it follows that every planar graph has a 5-colouring. □

7.5.4 4-Colour Theorem

All planar graphs are 4-colourable (Thm).

8 Matchings

8.1 Matching

A matching M of G is a set of edges s.t. each edge in M share no common vertices.

Vertices in M are saturated by the matching.

A maximum matching is the largest matching possible in G; a perfect matching is a type of maximum matching s.t. all vertices of G are saturated and $\therefore$ has size $\frac{n}{2}$.

An alternating path is a path s.t. consecutive edges alternate in M and not being in M; an augmenting path is a type of alternating path that starts and ends in distinct unsaturated vertices (of odd length).

8.2 Covers

A cover C of G is a set of vertices that have at least one endpoint of all edges in G.

In any graph, $|M| \leq |C|$ (Lem).

If $|M| = |C|$, then M is a maximum matching and C is a minimum cover (Lem).

Example. Say M is a maximum matching and C is a minimum cover of G. Show $|C| \leq 2|M|$.

Proof. Let $G = (V, E)$ be a graph such that M is the corresponding maximum matching and C is the corresponding minimum cover. Given M, there are $2|M|$ saturated vertices by a definition of a matching. Notice that the $2|M|$ saturated vertices of M forms a vertex cover C^* for G. Otherwise, if C^* is not a cover then there would exist an edge e that can then be added to M since e would not be incident to any saturated vertices in C^* which would contradict our assumption that M is a maximum matching. Moreover, C^* is not necessarily a minimum cover so

$$|C| \leq |C^*| = 2|M|$$

which proves the statement as desired. □

8.3 König's Theorem

In all bipartite graphs, the maximum size of a matching is the minimum size of a cover.

8.3.1 Bipartite Matching Algorithm: XY-Construction

We say that G has the cover $C = Y \cup (A \setminus X)$.

The XY-construction tries to keep finding augmenting paths and flipping them until there are no more.

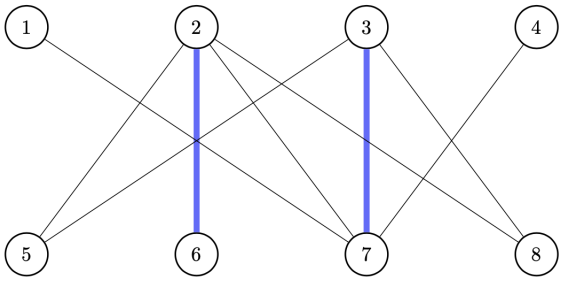
The current iteration stops and the next iteration begins when we find an unsaturated vertex in Y.

Halting Condition of XY-Construction:

(1) X_0 is empty.

(2) We cannot go from $X \to Y$.

Example. Apply the matching algorithm with M as an initial matching ($A = \{1, 2, 3, 4\}$).



Iteration 1:

$X_0 = \{1, 4\}$

$Y = \{7\}$; pr(7) = 1,4

$X = \{3\}$; pr(3) = 7

$Y = \{5, 8\}$; pr(5) = pr(8) = 3

5 is unsaturated. STOP.

Flip augmenting path:

5371

Iteration 2:

$X_0 = \{4\}$

$Y = \{7\}$; pr(7) = 4

$X = \{1\}$; pr(1) = 7

STUCK.

Now $Y \cup (A \setminus X) = \{2, 3, 7\}$ and $M = \{\{17\}, \{38\}, \{26\}\}$ so therefore

$$|M| = 3 = |Y \cup (A \setminus X)|$$

Tip:

- $X \to Y$
- $Y \rightsquigarrow X$

8.4 Application of Hall's Theorem

8.4.1 Hall's Theorem

Hall's Condition is satisfied if $\forall (D \subseteq A)$ that $|N(D)| \geq |D|$. If Hall's Condition holds, then by Hall's Theorem there exists a matching that saturates A.

If G is bipartite where Hall's Condition is satisfied and $|A| = |B|$ then G has a perfect matching.

Every k-regular graph has a perfect matching.

Example. Consider a standard deck of cards in a 4×13 array. Show that it is possible to pick one card from each column such that we get all 13 values.

Proof. Consider a graph G with bipartition (A, B). Let $A = \{A, 2, 3, \ldots, J, Q, K\}$ and $B = \{1, 2, \ldots, 11, 12, 13\}$. That is, A contains all 13 card values and B contains all of the columns where a vertex $x \in A$, $y \in B$ are adjacent if the card value x is in column y. To prove the statement it is sufficient to show that A has a perfect matching. Take a subset $D \subseteq B$ of k columns. Notice that there are $4k$ cards in these columns, and each value has at most 4 cards which implies that at least k values exist. So Hall's condition holds and there is therefore a perfect matching. □

8.5 Edge Colouring

An edge k-colouring of a graph G is an assignment of one of a set of k colours to each edge of G so that two edges incident with the same vertex are assigned different colours.

A bipartite graph with at most degree δ has an edge δ-colouring (*Thm.* 8.7.1).