1.
Combinatorial Analysis

1.1 Combinatorial Proof
1.2 Generating Series

Combinatorial Proof

(EX 1.1)
Examples of Counting Problems

Here are some examples of counting problems: How many ways can we

(a) put 6 objects in order?

(b) write a binary string of length 10 not containing 101?

(c) make change for \$1.00?

(d) deal a full house in poker?

We will use combinatorics and algebra to solve problems like this.

(EX 1.2)
Products and Sums

Consider the sets

$$A = \{a_1, a_2, a_3\}, B = \{b_1, b_2, b_3, b_4\}.$$

(a) *product*: The number of ways of choosing an element from A *and* an element from B is $|A||B| = 12$. This corresponds to the *Cartesian product* of A, B, denoted as $A \times B = \{(a,b) : a \in A \wedge b \in B\}$. For a finite sets A, B, the following equation holds.

Cartesian Product of Finite Sets

$$|A \times B| = |A||B|. \tag{1.1}$$

(b) *sum*: The number of ways of choosing an element from A *or* an element from B is $|A| + |B| = 7$. This corresponds to the *union* of A, B, denoted as $A \cup B = \{x : x \in A \vee x \in B\}$. Note that we have to be careful, since unions of finite sets correspond to sums only when they are disjoint.

Union of Disjoint Finite Sets

$$|A \cup B| = |A| + |B|. \tag{1.2}$$

(EX 1.3)
Subsets

Given a set S with $|S| = n$, how many subsets does S have?

Answer. First index the elements of S by $S = \{s_1, \ldots, s_n\}$.

- Note that, given any subset $T \subseteq S$, we can represent T by an n-tuple

  $$p = (p_1, \ldots, p_n) \in \{0,1\}^n$$

  whose entries are 0 or 1, where

  $$p_i = \begin{cases} 1 & \text{if } s_i \in T \\ 0 & \text{if } s_i \notin T \end{cases}$$

  for all $i \in \{1, \ldots, n\}$.

- Conversely, given any binary n-tuple $p = (p_1, \ldots, p_n)$, we can construct a subset $T \subseteq S$ by

  $$T = \{s_i \in S : p_i = 1\}.$$

This means that there is a bijection[1] between the collection of subsets of S, $\mathcal{P}(S)$, and the set of binary n-tuples, $\{0,1\}^n$. Thus

$$|\mathcal{P}(S)| = |\{0,1\}^n| \overset{[1.1]}{=} \prod_{i=1}^{n} |\{0,1\}| = 2^n.$$ ∎

[1] Recall that two sets are said to be *equinumerous* (i.e. the same size) if there is a bijection between them.

Therefore, we obtain the following general formula for any finite set S:

Number of Subsets

$$|\mathcal{P}(S)| = 2^{|S|}. \tag{1.3}$$

List of a Set

Def'n 1.1. Given a finite set S, a ***list*** of S is a $|S|$-tuple, which precisely has the elements of S.

(1.4) Lists

For any $S = \{s_1, s_2, s_3\}$, there are 6 lists of S.

$$(s_1, s_2, s_3), (s_1, s_3, s_2), (s_2, s_1, s_3), (s_2, s_3, s_1), (s_3, s_1, s_2), (s_3, s_2, s_1).$$

Note that, given any list $l = (l_1, l_2, l_3)$ of S, we have $4 - i$ choices for l_i. This idea can be generalized as follows.

Proposition 1.1. Lists

Let S be a set with $|S| = n$. Then there are $n(n-1)\cdots 1$ lists of S.

Proof. Since we are choosing $l_1 \in S, l_2 \in S \setminus \{l_1\}, \ldots, l_n \in S \setminus \{l_1, \ldots, l_{n-1}\}$, it follows that there are

$$|S \times S \setminus \{l_1\} \times \cdots \times S \setminus \{l_1, \ldots, l_{n-1}\}| \overset{[1.1]}{=} \prod_{i=1}^{n} \left| S \setminus \{l_j\}_{j=1}^{i} \right| = n(n-1)\cdots 1,$$

as desired. ∎

(1.5) Factorial

The number $n(n-1)\cdots 1$ is important, so we have a notation for it.

Factorial of a Nonnegative Integer

Def'n 1.2. Given $n \in \mathbb{N} \cup \{0\}$, we define the ***factorial*** of n, denoted as $n!$, by

$$n! = \prod_{k=1}^{n} k.$$

Partial List

Def'n 1.3. Let S be a set. A ***partial*** list of S is a k-tuple, where $k \leq |S|$, of distinct elements of S.

(1.6)

Observe that, given a set S with $|S| = n$ and $k \leq n$, to form a partial list $l = (l_1, \ldots, l_k)$, we are choosing $l_1 \in S, \ldots, l_k \in S \setminus \{l_1, \ldots, l_{k-1}\}$. Therefore, there are

$$|S \times S \setminus \{l_1\} \times \cdots \times S \setminus \{l_1, \ldots, l_{k-1}\}| \overset{[1.1]}{=} \prod_{i=1}^{k} n - k + i$$

partial lists of S of length k.

Partial Lists

$$|\{l : l \text{ is a length } k \text{ partial list of } S\}| = \prod_{i=1}^{k} |S| - k + i. \tag{1.4}$$

In case where $k \leq |S|$, note that we can simply write

$$\prod_{i=1}^{k} |S| - k + i = \frac{n!}{(n-k)!}. \tag{1.5}$$

Def'n 1.4. *n* **Choose** *k*
We write $\binom{n}{k}$, which reads n ***choose*** k to denote the number of k-element subsets of n-element set of S.

Proposition 1.2. *n* Choose *k*

Given $n \in \mathbb{N} \cup \{0\}, k \in \{0, \ldots, n\}$,

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

Proof. Let S be any set with $|S| = n$, and fix $k \in \{0, \ldots, n\}$. Recall that there are $\frac{n!}{(n-k)!}$ partial lists of S of length k. But note that choosing a length-k partial list of S is equivalent to choosing a k-element subset $X \subseteq S$ and then choosing a list of X. Since there are $k!$ lists of X, it follows that there are

$$\binom{n}{k} k!$$

length-k partial lists of S. But we assumed $k \leq n$, so from [1.4], [1.5],

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

is also the number of length-k partial lists of S, which means

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

or, equivalently

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

as required. ∎

(EX 1.7) Double Counting

The technique used in the proof of Proposition 1.2 is called *double counting*: it shows that two quantities are equal by showing that they represent two different ways of counting the same set. It is one of two combinatorial proofs. To demonstrate this technique once again, consider the following example. Prove that

$$\sum_k^n \binom{n}{k} = 2^n.$$

Proof. Let S be any set with $|S| = n$ and write

$$\mathcal{P}_k(S) = \{T \subseteq S : |T| = k\}$$

for all $k \in \{0, \ldots, n\}$. Now note that

$$\mathcal{P}(S) = \bigcup_{k=0}^{n} \mathcal{P}_k(S) \tag{1.6}$$

is a disjoint union. Thus

$$2^n \overset{[1.3]}{=} |\mathcal{P}(S)| \overset{[1.6]}{=} \left| \bigcup_{k=0}^{n} \mathcal{P}_k(S) \right| \overset{[1.2]}{=} \sum_{k=0}^{n} |\mathcal{P}_k(S)| \overset{\text{Def'n 1.4}}{=} \sum_{k=0}^{n} \binom{n}{k}.$$ ∎

(EX 1.8) Pascal's Identity

Let $n, k \in \mathbb{N}$. Then

Pascal's Identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}. \quad [1.7]$$

Proposition 1.3. Complement

Let $n \in \mathbb{N} \cup \{0\}$ and let $k \in \{0, \ldots, n\}$. Then

$$\binom{n}{k} = \binom{n}{n-k}. \quad [1.8]$$

(1.9) Multisets

Often times it is handy to allow duplicates for sets. Multisets are defined for this purpose.

Def'n 1.5. **Multiset**

A k-element ***multiset*** of n types is a n-tuple $(x_1, \ldots, x_n) \in (\mathbb{N} \cup \{0\})^n$ such that

$$\sum_{i=1}^{n} x_i = k.$$

Given $n, k \in \mathbb{N} \cup \{0\}$, we write $\left(\!\binom{n}{k}\!\right)$ to denote the number of k-element of multisets of n types.

Proposition 1.4.

Let $n, k \in \mathbb{N} \cup \{0\}$. Then

$$\left(\!\binom{n}{k}\!\right) = \binom{n+k-1}{n-1}. \quad [1.9]$$

Generating Series

(1.10) Formal Power Series

Before we begin to discuss about generating series, we are going to recall some properties of formal power series.

Recall 1.6. **Formal Power Series**

A ***formal power series*** is a formal sum[a] $\sum_{i=0}^{\infty} a_i x^i$, where we define the addition and multiplication analogous to those of polynomial.

(a) *addition*: Given $\sum_{i=0}^{\infty} a_i x^i, \sum_{i=0}^{\infty} b_i x^i$,

$$\sum_{i=0}^{\infty} a_i x^i + \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} (a_i + b_i) x^i.$$

(b) *multiplication*: Given $\sum_{i=0}^{\infty} a_i x^i, \sum_{j=0}^{\infty} b_j x^j$,

$$\left(\sum_{i=0}^{\infty} a_i x^i\right)\left(\sum_{j=0}^{\infty} b_j x^j\right) = \sum_{k=0}^{\infty} \sum_{i=0}^{k} a_i b_{k-i} x^k.$$

[a] A *formal sum* means that we are using summation symbolically. In fact, we typically think $\sum_{i_0}^{\infty} a_i x^i$ as an infinite sequence of coefficients, $(a_i)_{i=0}^{\infty}$.

(EX 1.11) Geometric Series

Let $P(x)=\sum_{i=0}^{\infty}x^i$ and consider multiplying $(1-x)$ to $P(x)$. Then we obtain

$$(1-x)P(x)=(1-x)\sum_{i=0}^{\infty}x^i=\sum_{i=0}^{\infty}x^i-\sum_{j=1}^{\infty}x^j=1.$$

Therefore, we write

Geometric Series

$$\frac{1}{1-x}=\sum_{i=0}^{\infty}x^i. \qquad [1.10]$$

Note that we are *not* concerned about convergence when we are dealing with formal power series. We rather focus on algebraic manipulations, as shown above.

Theorem 1.5. Binomial Theorem

Let $n\geq 0$. Then

$$(1+x)^n=\sum_{k=0}^{n}\binom{n}{k}x^k=\sum_{k=0}^{\infty}\binom{n}{k}x^k. \qquad [1.11]$$

Proof. We proceed inductively. Notice that the equation clearly holds when $n=0$. Now let $n\in\mathbb{N}$ and assume that

$$(1+x)^{n-1}=\sum_{k=0}^{n-1}\binom{n-1}{k}x^k \qquad [1.12]$$

(i.e. [1.11] is the inductive hypothesis). Then,

$$\begin{aligned}
(1+x)^{n-1}&=(1+x)(x+x)^{n-1}\\
&=(1+x)\sum_{k=0}^{n-1}\binom{n-1}{k}x^k && \text{by [1.12]}\\
&=\sum_{k=0}^{n-1}\binom{n-1}{k}x^k+\sum_{k=0}^{n-1}\binom{n-1}{k}x^{k+1}\\
&=\sum_{k=0}^{n-1}\binom{n-1}{k}x^k+\sum_{k=1}^{n}\binom{n-1}{k-1}x^k\\
&=\binom{n-1}{0}x^0+\sum_{k=1}^{n-1}\left(\binom{n-1}{k}+\binom{n-1}{k-1}\right)x^k+\binom{n}{n}x^n\\
&=\binom{n}{0}x^0+\sum_{k=1}^{n-1}\binom{n}{k}x^k+\binom{n}{n}x^n && \text{by [1.7]}\\
&=\sum_{k=0}^{n}\binom{n}{k}x^k,
\end{aligned}$$

as desired. The second equality follows from the fact that $\binom{n}{k}=0$ whenever $n<k$. ∎

Def'n 1.7. **Inverse** of a Formal Power Series

Let $F(x)$ be a formal power series. If there exists a formal power series $G(x)$ such that $F(x)G(x)=1$, then we call G the *inverse* of $F(x)$, denoted as $F(x)^{-1}$.[a]

[a] Of course, we can say G is *the* inverse of F and write $G=F^{-1}$ because an inverse of a formal power series is unique if exists.

Proposition 1.6. Invertible Formal Power Series

Let $F(x) = \sum_{k=0}^{\infty} a_k x^k$. Then F has the inverse if and only if $a_0 \neq 0$.

Theorem 1.7.
Negative Binomial Theorem

Let $n \in \mathbb{N}$. Then

$$(1-x)^{-n} = \sum_{k=0}^{\infty} \binom{n+k-1}{n-1} x^k. \tag{1.13}$$

Proof. We proceed combinatorially. We begin by denoting

$$\mathcal{M} = (\mathbb{N} \cup \{0\})^n, \mathcal{M}_k = \{M \in \mathcal{M} : |M| = k\}.$$

Then notice that $\mathcal{M} = \bigcup_{k=0}^{\infty} \mathcal{M}_k$ is a disjoint union, so

$$\begin{aligned}
\sum_{k=0}^{\infty} \binom{n+k-1}{n-1} x^k &= \sum_{k=0}^{\infty} |\mathcal{M}_k| x^k && \text{by [1.9]} \\
&= \sum_{m \in \mathcal{M}} x^{|M|} \\
&= \sum_{k_1, \ldots, k_n \in \mathbb{N} \cup \{0\}} x^{\sum_{i=1}^{n} k_i} \\
&= \sum_{k_1=0}^{\infty} \cdots \sum_{k_n=0}^{\infty} x^{k_1} \cdots x^{k_n} \\
&= \sum_{k_1=0}^{\infty} x^{k_1} \cdots \sum_{k_n=0}^{\infty} x^{k_n} \\
&= \left(\sum_{k=0}^{\infty} x^k\right)^n \\
&= \left(\frac{1}{1-x}\right)^n && \text{by [1.10]} \\
&= (1-x)^{-n},
\end{aligned}$$

as desired. ∎

(EX 1.12) Write $\left(\frac{x}{1+2x}\right)^5$ as a formal power series.

Answer. Observe that

$$\begin{aligned}
\left(\frac{x}{1+2x}\right)^5 &= x^5 (1-(-2x))^{-5} \\
&= x^5 \sum_{k=0}^{\infty} \binom{k+5-1}{5-1} (-2x)^k && \text{by Theorem 1.7} \\
&= \sum_{k=0}^{\infty} \binom{n+4}{4} (-2)^k x^{k+5} \\
&= \sum_{k=5}^{\infty} \binom{n-1}{4} (-2)^{n-5} x^n,
\end{aligned}$$

is the formal power series that we are looking for. ∎

(1.13)
Coefficeint Extraction

Since we are treating formal power series *formally*, we are more concerned about coefficients. For this reason, we introduce the following notation that *extracts* out the coefficient of interest.

Notation 1.8. **Coefficient Extraction** of a Formal Power Series
Given a forma power series $F(x) = \sum_{k \geq 0}^{\infty} a_n x^n$, we write $\left[x^k\right] F(x)$ to denote a_k, the coefficient of x^k in $F(x)$.

The following equailties follow immediately from Notation 1.8: given any formal power series $F(x), G(x)$,

(a) *addition*: $\left[x^k\right](aF(x)+bG(x)) = a\left[x^k\right]F(x)+b\left[x^k\right]G(x)$; and

(b) multiplication: $\left[x^k\right](F(x)G(x)) = \sum_{l=0}^{k}\left[x^l\right]F(x)\left[x^{k-l}\right]G(x)$.

A special case of (b) is when $G(x)=x^l$ for some $l\in\mathbb{N}\cup\{0\}$: we obtain $\left[x^k\right]x^lF(x)=\left[x^{k-l}\right]F(x)$.

(1.14) Weight Functions

We now begin to discuss about generating series. Generating series are special in a sense that their coefficients are determined by some *weight* function.

Def'n 1.9. **Weight** Function

Let S be a set. We say a function of the form $w:S\to\mathbb{N}\cup\{0\}$ a ***weight*** function on S if for all $k\in\mathbb{N}\cup\{0\}$, there are finitely many $s\in S$ such that $w(s)=k$.

And we define the generating series for S as follows.

Def'n 1.10. **Generating Series** for a Set

Let S be a set and let $w:S\to\mathbb{N}\cup\{0\}$ be a weight function on S. Then we define the ***generating series*** for S with respect to w, denoted as $\Phi_{S,w}(x)$, by

$$\Phi_{S,w}(x)=\sum_{s\in S}x^{w(s)}.$$

Often times the weight function that we are working with is understood, and for such cases, we simply write $\Phi_S(x)$ instead for simplicity.

Proposition 1.8. *Let S be a set and let w be a weight function on S. Then*

$$\Phi_S(x)=\sum_{k=0}^{\infty}|\{s\in S: w(s)=k\}|x^k. \quad [1.14]$$

Proof. Let $S_k=\{s\in S: w(s)=k\}$ for all $k\in\mathbb{N}$, where we also assume that each S_k is finite. Then,

$$\Phi_S(x)=\sum_{s\in S}x^{w(s)}=\sum_{k=0}^{\infty}\left(\sum_{s\in S_k}x^k\right)=\sum_{k=0}^{\infty}|S_k|x^k.$$ ∎

Corollary 1.8.1. *Consider the setting of Proposition 1.8. For each $k\in\mathbb{N}\cup\{0\}$,*

$$\left[x^k\right]\Phi_S(x)=|S_k|.$$

(EX 1.15) Binary Strings

Let $S=\{\text{binary string}\}$ and let $w:S\to\mathbb{N}\cup\{0\}$ by

$$w(s)=\text{length of } s.$$

Then the generating series for S is

$$\Phi_S \overset{[1.14]}{=} \sum_{k=0}^{\infty}(\text{number of binary strings of length } k)\,x^k=\sum_{k=0}^{\infty}2^kx^k \overset{[1.10]}{=} \frac{1}{1-2x}.$$

(EX 1.16) Subsets

Let $S=\mathcal{P}(\{1,\dots,n\})$ (write $[n]=\{1,\dots,n\}$ for convenience) and let $w:S\to\mathbb{N}\cup\{0\}$ by

$$w(s)=|s|$$

for all $s \in S$. Then the generating series of S is

$$\Phi_S(s) \overset{[1.14]}{=} \sum_{k=0}^{\infty} (\text{number of } k\text{-element subsets of } [n])\, x^k \overset{\text{Def'n 1.4}}{=} \sum_{k=0}^{\infty} \binom{n}{k} x^k \overset{\text{Theorem 1.5}}{=} (1+x)^n.$$

Theorem 1.9.
Sum Lemma

Let S_1, S_2 be disjoint sets and let $w : S \to \mathbb{N} \cup \{0\}$ be a weight function on $S = S_1 \cup S_2$. Then

$$\Phi_{S_1}(x) + \Phi_{S_2}(x) = \Phi_{S_1 \cup S_2}(x), \tag{1.15}$$

where each $\Phi_{S_i}(x)$ is with respect to the restriction $w|_{S_i} : S_i \to \mathbb{N} \cup \{0\}$.

Proof. Observe that

$$\Phi_{S_1}(x) + \Phi_{S_2}(x) = \sum_{s \in S_1} x^{w(s)} + \sum_{s \in S_2} x^{w(s)} = \sum_{s \in S_1 \cup S_2} x^{w(s)} = \Phi_{S_1 \cup S_2}(x)$$

since S_1, S_2 are disjoint. ∎

Theorem 1.10.
Infinite Sum Lemma

Let $S_0, S_1, \dots$ be disjoint and let $S = \bigcup_{k=0}^{\infty} S_k$. If $w : S \to \mathbb{N} \cup \{0\}$ is a weight function on S, then

$$\Phi_S(x) = \sum_{k=0}^{\infty} \Phi_{S_k}(x), \tag{1.16}$$

where each $\Phi_{S_k}(x)$ is with respect to $w|_{S_k} : S_k \to \mathbb{N} \cup \{0\}$.

Proof. Similar to the presented proof of Theorem 1.9. ∎

Theorem 1.11.
Product Lemma

Let S_1, S_2 be sets and let w_1, w_2 be weight functions on S_1, S_2, respectively. Then

$$\Phi_{S_1}(x)\, \Phi_{S_2}(x) = \Phi_{S_1 \times S_2}(x) \tag{1.17}$$

where $\Phi_{S_1 \times S_2}(x)$ is with respect to $w : S_1 \times S_2 \to \mathbb{N} \cup \{0\}$ by

$$w(s_1, s_2) = w_1(s_1) + w_2(s_2)$$

for all $(s_1, s_2) \in S_1 \times S_2$.

Proof. Note that

$$\begin{aligned}
\Phi_{S_1}(x)\, \Phi_{S_2}(x) &= \sum_{s_1 \in S_1} x^{w_1(s_1)} \sum_{s_2 \in S_2} x^{w_2(s_2)} \\
&= \sum_{s_1 \in S_1} \sum_{s_2 \in S_2} x^{w_1(s_1) + w_2(s_2)} = \sum_{(s_1, s_2) \in S_1 \times S_2} x^{w(s_1, s_2)} = \Phi_{S_1 \times S_2}(x),
\end{aligned}$$

as desired. ∎

2. Strings

2.1 Strings
2.2 Binary Strings

Strings

Notation 2.1. S^*

Let S be a set. We write S^* to denote

$$S^* = \bigcup_{p=0}^{\infty} S^p,$$

the set of every tuple of the form $(s_1, \ldots, s_p) \in S^p$ for some $p \in \mathbb{N} \cup \{0\}$.[a]

[a] Note that we write S^0 to denote $\{()\}$ and S^1 to denote $\{(s) : s \in S\}$.

(EX 2.1)

(a) $\{0,1\}^* = \{(), (0), (1), (0,0), (0,1), (1,0), (1,1), (0,0,0), \ldots\}$. Note that we can also think each element of $\{0,1\}^*$ as a *binary string*, $\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots$.[1]

(b) $\mathbb{N}^* = \{(), (1), (2), \ldots, (1,1), (1,2), \ldots, (1,1,1), \ldots\}$.

(2.2) Weight Function on S^*

Let S be a set and suppose that we have a weight function $w : S \to \mathbb{N} \cup \{0\}$ on S. If there does not exist $s \in S$ such that $w(s) = 0$, then we define $w^* : S^* \to \mathbb{N} \cup \{0\}$ by

$$w^*((s_1, \ldots, s_k)) = \sum_{i=1}^{k} w(s_i). \qquad [2.1]$$

Note that we have to ensure that no element of S has weight 0, since otherwise we obtain infinitely many elements in S^* that has weight 0.

(EX 2.3) If $S = \{0,1\}$ and $w(0) = w(1) = 1$, then w^* is the *length* function.

Theorem 2.1. String Lemma

Let S be a set with a weight function $w : S \to \mathbb{N} \cup \{0\}$, with no $s \in S$ such that $w(s) = 0$. Then

$$\Phi_{S^*}(x) = \frac{1}{1 - \Phi_S(x)} \qquad [2.2]$$

where $\Phi_{S^*}(x)$ is with respect to $w^* : S^* \to \mathbb{N} \cup \{0\}$.

Proof. Recall that $S^* = \bigcup_{k=0}^{\infty} S^k$ is a disjoint union, so

$$\begin{aligned}
\Phi_{S^*}(x) &= \sum_{k=0}^{\infty} \Phi_{S^k}(x) && \text{by the infinite sum lemma} \\
&= \sum_{k=0}^{\infty} (\Phi_S(x))^k && \text{by the product lemma} \\
&= \frac{1}{1 - \Phi_S(x)}, && \text{by [1.10]}
\end{aligned}$$

as desired. ∎

(EX 2.4) Let $S = \{0,1\}$ and let $w(0) = w(1) = 1$. Find $\Phi_{S^*}(x)$.

Answer. Notice that

$$\Phi_S(x) = 1x^0 + 2x^1 + 4x^2 + \cdots = \sum_{k=0}^{\infty} (2x)^k,$$

[1] ε represents the empty string.

so by Proposition 2.1,

$$\Phi_{S^*}(x) = \frac{1}{1-\Phi_S(x)} = \frac{1}{1-2x}.$$

Def'n 2.2. **Composition** of k Parts

A ***composition*** of k parts is a k-tuple $(n_1,\ldots,n_k)$ with $k \in \mathbb{N}\cup\{0\}$, $n_1,\ldots,n_k > 0$. Each n_i is called a ***part*** of $(n_1,\ldots,n_k)$. The ***weight*** of $(n_1,\ldots,n_k)$ is the sum $\sum_{i=1}^{k} n_i$.[a]

[a] We often say a *composition of n* to mean a composition with weight n.

(2.5) Note that it is equivalent to define a composition to be an element of $\mathbb{N}^*$, where the weight is given by $w^*: \mathbb{N}^* \to \mathbb{N}\cup\{0\}$ where $w(m) = m$ for all $m \in \mathbb{N}$. This means that, if we let

$$S = \{\text{compositions}\} = \mathbb{N}^*$$

then

$$\Phi_S(x) = \Phi_{\mathbb{N}^*}(x) \overset{[1.15]}{=} \frac{1}{1-\sum_{k=0}^{\infty} x^n} = \frac{1}{1-x\sum_{k=0}^{\infty} x^k} \overset{[1.10]}{=} \frac{1}{1-\frac{x}{1-x}} = \frac{1-x}{1-2x}.$$

Generating Series of the Collection of Compositions

$$\Phi_{\mathbb{N}^*}(x) = \frac{1-x}{1-2x}. \qquad [2.3]$$

[2.3] allows us to count compositions by computing coefficients: the number of compositions of $n \in \mathbb{N}$ is given by

$$[x^n]\,\Phi_{\mathbb{N}^*}(x) = [x^n]\,(1-x)\sum_{n=0}^{\infty}(2x)^n.$$

(EX 2.6) How many compositions are there of 239 where all parts are odd?

Answer. Note that if we let

$$T = \{1,3,\ldots\} = \{k \in \mathbb{N} : k \bmod 2 \equiv 1\}$$

then T^* is the set of compositions whose parts are odd. Now note that

$$\Phi_T(x) = \sum_{k\in\mathbb{N}:k \bmod 2\equiv 1} x^n = x\sum_{k=0}^{\infty} x^{2k} \overset{[1.10]}{=} \frac{x}{1-x^2},$$

so the number of compositions of 239 with odd parts is the coefficient of x^{239} in $\Phi_{T^*}(x)$, which can be computed as

$$\begin{aligned}
\left[x^{239}\right]\Phi_{T^*}(x) &= \left[x^{239}\right]\frac{1}{1-\Phi_T(x)} && \text{by the string lemma}\\
&= \left[x^{239}\right]\frac{1}{1-\frac{x}{1-x^2}}\\
&= \left[x^{239}\right]\frac{1-x^2}{1-x-x^2}.
\end{aligned}$$

But unfortunately, with the tools that we developed so far, we cannot evaluate $\left[x^{239}\right]\frac{1-x^2}{1-x-x^2}$. To do so, first write down $\frac{1-x^2}{1-x-x^2}$ as a power series:

$$\frac{1-x^2}{1-x-x^2} = \sum_{k=0}^{\infty} a_k x^k \qquad [2.4]$$

for some $a_0, a_1, \ldots$. Note that we can get rid of the denominator in the LHS of [2.4] by multiplying $1-x-x^2$ to both sides of [2.4]:

$$1-x^2 = \left(1-x-x^2\right)\sum_{k=0}^{\infty} a_k x^k. \tag{2.5}$$

We then can clean up the RHS of [2.5] by grouping like terms:

$$1-x^2 = a_0 + (a_1-a_0)\,x + \sum_{k=2}^{\infty}\left(a_k - a_{k-1} - a_{k-2}\right)x^k. \tag{2.6}$$

But two formal power series are said to be equal if and only if they have the same coefficients. This means we get a system of *infinitely many* equation, so to speak, from [2.6]:

$$\begin{cases} 1 & = a_0 \\ 0 & = a_1 - a_0 \\ -1 & = a_2 - a_1 - a_0 \\ 0 & = a_3 - a_2 - a_1 \\ & \vdots \end{cases} \tag{2.7}$$

is the system. Although [2.7] involves infinitely many equations, we can still solve it. First note that the first three equations form a system of three equations with three indeterminates, whose solution is $a_0 = a_1 = a_2 = 1$. Moreover, by rearranging

$$0 = a_k - a_{k-1} - a_{k-2}$$

which holds for all $k \geq 3$, we obtain that

$$a_k = a_{k-1} + a_{k-2}, \tag{2.8}$$

a recurrence relation that defines the Fibonacci sequence. Therefore, the coefficients are:

$$(a_0, a_1, a_2, a_3, a_4, a_5, \ldots) = (1, 1, 1, 2, 3, 5, 8, \ldots),$$

and we can use the recurrence relation [2.8] to find, in particular, $\left[x^{239}\right]\Phi_{T^*}(x) = a_{239}$. Although this does not (yet) give the explicit formula for a_{239}, we shall see later how to obtain a formula for a problem like this. ∎

(EX 2.7) Let b_k be the number of compositions of k with an odd number of parts. Find

(a) an expression for $\sum_{k=0}^{\infty} b_k x^k$ as a rational function; and

(b) a formula for b_n.

Answer. Let

$$S = \{\text{compositions with an odd number of parts}\} = \bigcup_{k\in\mathbb{N}\cup\{0\}}^{\infty} \mathbb{N}^{2k+1},$$

so the generating series $\Phi_S(x)$ for S (i.e. $\sum_{k=0}^{\infty} b_k x^k$) is given by

$$\begin{aligned}
\sum_{k=0}^{\infty} &= \Phi_S(x) \\
&= \sum_{k=0}^{\infty} \Phi_{\mathbb{N}^{2k+1}} && \text{by the infinite sum lemma} \\
&= \sum_{k=0}^{\infty} \left(\Phi_{\mathbb{N}}(x)\right)^{2k+1} && \text{by the product lemma} \\
&= \sum_{k=0}^{\infty} \left(\frac{x}{1-x}\right)^{2k+1} && \text{by the geometric series} \\
&= \frac{\frac{x}{1-x}}{1-\left(\frac{x}{1-x}\right)^2} && \text{by the geometric series} \\
&= \frac{x(1-x)}{1-2x} && [2.9] \\
&= x(1-x)\sum_{k=0}^{\infty} (2x)^k && \text{by the geometric series} \\
&= x + \sum_{k=2}^{\infty} 2^{k-2} x^k,
\end{aligned}$$

which means

$$b_k = \begin{cases} 0 & \text{if } k = 0 \\ 1 & \text{if } k = 1 \\ 2^{k-2} & \text{if } k \geq 2 \end{cases}. \qquad [2.10]$$

Note that [2.9] is the answer for (a) and [2.10] is the answer for (b). ∎

(2.8) In summary, so far we looked at the applications of the sum, product, string lemmas to describe the number of compositions of n with restrictions on

(a) the number of parts; and / or

(b) what tha parts look like.

They (the lemmas) give the generating series $\sum_{k=0}^{\infty} a_k x^k$ as a rational function, and a formula or recurrence relation for a_k.

Binary Strings

Def'n 2.3. **Binary String**

A ***binary string*** s is an expression of the form

$$s = b_1 b_2 \dots b_t,$$

where $t \in \mathbb{N} \cup \{0\}$ and $b_1, \dots, b_k \in \{0,1\}$. Each b_i is called a ***bit***, and k is called the length of s. The ***empty string*** is denoted by ε.

(2.9) We can identify binary strings with elements of $\{0,1\}^*$: note that

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

whereas

$$\{\text{binary strings}\} = \{\varepsilon, 0, 1, 00, \dots\}.$$

For this reason, counting problems related to binary strings can be translated to one that involves counting the elements of $\{0,1\}^*$. For instance,

$$(\text{number of binary strings of length } n) = [x^n]\,\Phi_{\{0,1\}^*}(x) = \cdots = [x^n]\,\frac{1}{1-2x} \overset{[1.10]}{=} 2^n,$$

where $\Phi_{\{0,1\}^*}(x)$ is with respect to w^* with $w(0) = w(1) = 1$. We want to apply ideas from generating series to count strinngs in various sets. For instance, consider the following questions:

(a) How many binary strinsg of length 12 are there not containing 101 as a substring?

(b) How many binary strings of length 20 are there where every 0 is followed by a 1?

We desire to express these questions in terms of certain sets on which we can apply the sum, product, string lemmas. First we introduce few definitions.

Def'n 2.4. **Concatination** of Strings

Let $s_1 = a_1 \ldots a_n$ and $s_2 = b_1 \ldots b_m$. Then the ***concatination*** of s_1, s_2, denoted by s_1s_2, is the binary string

$$s_1s_2 = a_1 \ldots a_n b_1 \ldots b_m.$$

We write s_1^k to denote the concatination of k s_1's,

$$s_1^k = \underbrace{s_1 \cdots s_1}_{k\ s_1\text{'s}}.$$

Def'n 2.5. **Substring** of a String

Let s be a string. s' is a substring of s if there exists strings t_1, t_2 such that

$$s = t_1 s' t_2.$$

Def'n 2.6. **Concatination Product** of Sets of Strings

Let S_1, S_2 be sets of strings. We define the ***concatination product*** of S_1, S_2, denoted as S_1S_2, by

$$S_1S_2 = \{s_1s_2 : s_1 \in S_1, s_2 \in S_2\}.$$

We write $S_1^{(k)}$ to denote[a]

$$S_1^{(k)} = \underbrace{S_1 \cdots S_1}_{k\ S_1\text{'s}}.$$

[a]Note that the exponent is of the form $S^{(k)}$ but not S^k. This notation is necessary to avoid confusion with the Cartesian product of a set with itself.

Def'n 2.7. **Regular Expression**

A ***regular expression*** is defined recursively: a regular expression is

(a) 0, 1, ε;

(b) an expression $R_1 \cup R_2$, where R_1, R_2 are regular expressions;

(c) an expression R_1R_2, where R_1, R_2 are regular expressions; or

(d) an expressino R^*, where R is a regular expression.

(EX 2.10)
Examples of Regular Expressions

The following are regular expressions:

(a) 110;

(b) $110 \cup 10$;

(c) $(110 \cup 10)^*$;

(d) $(01)(110 \cup 10)^*(1^* \cup \varepsilon)$; and

(e) $(11)^5 \cup \big((01)(110 \cup 10)^*(1^* \cup \varepsilon)\big)$.

Def'n 2.8. **Production** of a Regular Expression

We define ***produce*** recursively:

(a) $\varepsilon, 0, 1$ produce the sets $\{\varepsilon\}, \{0\}, \{1\}$, respectively.

(b) If R_1, R_2 produce $\mathcal{R}_1, \mathcal{R}_2$ respectively, then:

  (i) $R_1 \cup R_2$ produces $\mathcal{R}_1 \cup \mathcal{R}_2$;

  (ii) $R_1 R_2$ produces $\mathcal{R}_1 \mathcal{R}_2$; and

  (iii) R_1^* produces $\bigcup_{k=0}^{\infty} \mathcal{R}_1^{(k)}$.

(EX 2.11)
Examples of Production

(a) $R_1 = \varepsilon \cup 0 \cup 1$ produces $\mathcal{R} = \{\varepsilon, 0, 1\}$.

(b) $R_2 = (00)(1 \cup 01)$ produces $\mathcal{R}_2 = \{001, 0001\}$.

(c) $R_3 = (10)^5(111)$ produces $\mathcal{R}_3 = \{1010101010111\}$.

(d) $R_4 = (01) \cup (10)^*$ produces $\mathcal{R}_4 = \{01, \varepsilon, 10, 1010, 101010, \ldots\}$.

(d) in particular shows that a finite regular expression can produce an infinite set of binary strings.

(EX 2.12)
Producing Binary Strings

Notice that $(0 \cup 1)^*$ produces $\{\text{binary strings}\}$.

Def'n 2.9. **Rational Language**

If $\mathcal{R}$ is a set of strings produced by a regular expression R, then $\mathcal{R}$ is a ***rational language***.

This page intentionally left blank.

3. Graph Theory

3.1 Graphs
3.2 Subgraphs
3.3 Eulerian Circuits

Graphs

Def'n 3.1. **Undirected Graph**
An ***undirected graph*** is an ordered pair of sets (V,E) where V is nonempty and E is a set of unordered pairs of elements of V. The elements of V are called ***vertices*** and the elements of E are called edges.

(3.1) Since we are going to consider undirected graphs only, we are going to say $G=(V,E)$ is a *graph* to mean that G is an *undirected graph*. We introduce some terminologies and notations regarding graphs that we are going to use. Let $G=(V,E)$ be a graph.

Def'n 3.2. **Adjacent** Vertices
We say $u,v\in V$ are ***adjacent*** if $\{u,v\}\in E$.

Def'n 3.3. **Neighbor, Neighborhood** of a Vertex
Let $v\in V$. A ***neighbor*** of v is a vertex adjacent to v, and the set of the neighbors of v is called the ***neighborhood*** of v, denoted as $N_G(v)$.

Given an edge $e=\{u,v\}\in E$, we say e is ***incident*** with u,v. Alternatively, we say e ***joins*** u,v.

(a) Given $e=\{u,v\}\in E$, we will often write $e=uv$ instead. Since edges are unordered, $e=uv=vu$.

(b) We are going to consider ***simple*** graphs only. That is, we are going to avoid ***loops*** (i.e. edges of the form vv for some $v\in V$) and ***multiple edges*** (i.e. there is at most one edge uv for all $u,v\in V, u\neq v$; this is evident from our definition). Graphs that allow loops and multiple edges are called ***multigraphs***.

(c) We are going to consider ***finite*** graphs only (i.e. V is finite).

(d) Given a graph G, we shall write $V(G), E(G)$ to denote the set of vertices and the set of edges, respectively.

Def'n 3.4. **Degree** of a Vertex
The ***degree*** of $v\in V$, denoted as $\deg(v)$, is the size of its neighborhood: $\deg(v)=|N_G(v)|$.

Theorem 3.1. Handshaking Lemma

For every graph G

$$\sum_{v\in V(G)}\deg(v)=2|E(G)|.$$

Proof. Note that each edge $uv\in E(G)$ contributes 2 to the sum, one for $\deg(u)$, one for $\deg(v)$. ∎

Corollary 3.1.1.

For any graph G, the number of vertices of odd degree is even.

Proof. Let $\mathcal{O},\mathcal{E}\subseteq V(G)$ be the set of vertices of odd degree and the set of vertices of even degree, respectively. Then $\mathcal{O}\cup\mathcal{E}=V(G)$ is a disjoint union, so

$$\sum_{v\in V(G)}\deg(v)=\sum_{v\in\mathcal{O}}\deg(v)+\sum_{v\in\mathcal{E}}\deg(v).$$

But the first sum in the above equality is even by the handshaking lemma and the last sum is clearly even. Thus $\sum_{v\in\mathcal{O}}\deg(v)$ is even as well. ∎

Corollary 3.1.2.
Average Vertex Degree

The average of vertex degree of any graph G is $\frac{2|E(G)|}{|V|}$.

(3.2)
Graph Isomorphisms

We consider an important equivalence releation for graphs.

Def'n 3.5. **Isomorphic** Graphs
Let G_1, G_2 be graphs. We say G_1, G_2 are ***isomorphic*** if there exists a bijection $\phi : V(G_1) \to V(G_2)$, called an ***isomorphism***, such that $uv \in E(G_1)$ if and only if $f(u)f(v) \in E(G_2)$.

In other words, a graph isomorphism preserves two things: the vertex set and the adjacency of the vertices.

(3.3)
Isomorphism Classes

Graph isomorphism is an equivalence relation. That is, given graphs G_1, G_2, G_3,

(a) *reflexive*: every graph is isomorphic to itself;

(b) *symmetric*: if G_1 is isomorphic to G_2, then G_2 is isomorphic to G_1; and

(c) *transitive*: if G_1 is isomorphic to G_2 and G_2 is isomorphic to G_3, then G_1 is isomorphic to G_3.

All graphs that are isomorphic to each other form an ***isomorphism class*** of graphs. Any graph of an isomorphism class can be chosen as a *representative* for the whole class.

Def'n 3.6. **Complete** Graph
A ***complete graph*** is a graph every pair of vertices is an edge. A complete graph on n vertices is denoted as K_n.

Def'n 3.7. **Regular** Graph
Given $k \in \mathbb{N} \cup \{0\}$, we say a graph G is ***k-regular*** if $\deg(v) = k$ for all $v \in V(G)$.

(3.4)
Graph Families

We have special names for $k = 0, 1, 2$.

(a) A 0-regular graph is called *isolated vertices*.

(b) A 1-regular graph is called *matchings*.

(c) A 2-regular graph is called *cycles*.

Also, every $(|V(G)| - 1)$-regular graph is *complete*. The number of edges of a k-regular graph is $\frac{|V|k}{2}$.

Def'n 3.8. **Bipartite** Graph
A graph G is ***bipartite*** if there exists a partition $\{A, B\}$ of $V(G)$ such that every $e \in E(G)$ joins a vertex in A and a vertex in B.

The smallest non-bipartite graph is K_3.

Def'n 3.9. **Complete Bipartite** Graph
A graph G is called ***complete bipartite*** if there exists a partition $\{A, B\}$ of $V(G)$ such that for every $v \in A, u \in B$, $uv \in E(G)$. We write $G = K_{m,n}$ where $|A| = m, |B| = n$.

Note that $E(K_{m,n}) = mn$.

Def'n 3.10. ***n*-cube**

Given $n \in \mathbb{N}$, the ***n*-cube** G is a graph such that

$$V(G) = \{0,1\}^n,$$

the set of length n binary strings, and for every $u, v \in V(G)$, $uv \in E(G)$ if and only if u, v differ in exactly one position.

(3.5)
n-cubes

Let $n \in \mathbb{N}$ and let G be the n-cube.

(a) $|V(G)| = 2^n$, which is the number of binary strings of length n.

(b) G is n-regular, since given any vertex $v \in V(G)$, we obtain a neighbor of v by changing one of n bits.

(c) Since there are 2^n vertices each with degree n, the sum of degrees is $n2^n$, so by the handshaking lemma, $|E(g)| = n2^{n-1}$.

(d) G is bipartite.

Proof. Let

$$\begin{aligned} A &= \{v \in V(G) : v \text{ has odd number of 0's}\} \\ B &= \{v \in V(G) : v \text{ has even number of 0's}\} \end{aligned}.$$

Then $\{A, B\}$ is a partition of V clearly. ◁

(3.6)
Recursive Construction of *n*-cubes

(a) Make 2 copies of the $(n-1)$-cube.

(b) Add 0 in front of every vertex in one copy, 1 for the another copy.

(c) Join corresponding copies of the verties (i.e. verties who differ by only the first bit).

Subgraphs

Def'n 3.11. **Walk, Path** of a Graph

Let $G = (V, E)$ be a graph.

(a) A u,v***-walk*** of G is a sequence $(v_i)_{i=0}^k$ of vertices such that $v_0 = u$, $v_k = v$, and $v_{i-1}v_i \in E$ for all $i \in \{1, \ldots, k\}$. The number of vertices (which is k for this case) is called the ***length*** of the walk. We say a walk is ***closed*** if $v_0 = v_k$.

(b) A u,v***-walk*** is a u,v-walk with no repeated vertices.

Proposition 3.2. *Let G be a graph and let $u, v \in V(g)$. If there exists a u,v-walk in G, then there is a u,v-path in G.*

Proof. We proceed inductively (strong induction). Let $(v_i)_{i=0}^k$ be a $u - v$ walk in G.

- The result is clear when $(v_i)_{i=0}^k$ does not have any repeated vertices. ◁

- Suppose that the result holds for $u-v$ walks with less than $n \in \mathbb{N}$ repeated vertices, and let $(v_i)_{i=0}^k$ be a $u-v$ walk with n repeated vertices. Let i_0, i_1 be indices such that $v_{i_0} = v_{i_1}$. Then by removing $v_{i_0+1}, \ldots, v_{i_1}$, the number of repetitions reduces by at least 1. Thus by the induction hypothesis, the result holds. ∎

Corollary 3.2.1. *Let G be a graph and let $u, v, w \in V(G)$. If there exists a u,v-path and v,w-path in G, then there is a u,w-path in G.*

Proof. By joining v,w-path at the end of a u,v-path, we obtain a u,w-path. The result follows from Proposition 3.2 then. ∎

Def'n 3.12. **Subgraph** of a Graph
Let G be a graph. We say a graph H is a ***subgraph*** of G if $V(H) \subseteq V(G), E(H) \subseteq E(G)$. If $V(H) = V(G)$, we say H is ***spanning***. If $H \neq G$, we say H is ***proper***.

(3.7) For notational ease,

(a) we write $H - e$ for some $e \in E(H)$ to denote the subgraph with vertex set $V(H)$ and edge set $E(H) \setminus \{e\}$; and

(b) we write $H + e$ to denote the graph with vertex set $V(H)$ and edge set $E(H) \cup \{e\}$.

Def'n 3.13. **Cycle** of a Graph
A ***cycle*** in a graph G is a subgraph with $k \in \mathbb{N}$ distinct vertices $v_1, \ldots, v_k \in V(G)$ and k distinct edges $v_1v_2, \ldots, v_{k-1}v_k, v_kv_1$. Such a cycle has ***length*** k.

(3.8) The following results directly follow from Def'n 3.13.

(a) The minimum length of a cycle is 3.

(b) We can represent a cycle as a closed walk.

(c) If uv is an edge of a cycle C, then $C - uv$ is a u,v-path (if we consider paths as subgraphs). On the other hand, if P is a uv-path, then $P + uv$ is a cycle.

(d) A cycle is a 2-regular graph.

Proposition 3.3. *Let G be a graph. If $\deg(v) \geq 2$ for all $v \in V(G)$, then G has a cycle.*

Proof. Let $(v_i)_{i=1}^k$ be a longest path in G. Since $\deg(v_0) \geq 2$, v_0 has a neighbor distinct from v_1, say u. But by the maximality of $(v_i)_{i=1}^k$, there exists $i \in \{2, \ldots, k\}$ such that $v_i = u$. It follows that H with $V(H) = \{v_1, \ldots, v_i\}, E(H) = \{v_1v_2, \ldots, v_{i-1}v_i, v_iv_1\}$ is a cycle of G. ∎

Def'n 3.14. **Girth** of a Graph
Let G be a graph. If G has a cycle, then the ***girth*** of G is the length of the longest cycle of G. Otherwise, the girth of G is ∞.

Def'n 3.15. **Hamilton** Cycle
Let G be a graph. A ***Hamilton*** cycle of G is a spanning cycle.

Def'n 3.16. **Connected** Graph
Let G be a graph. We say G is ***connected*** if for every $u, v \in V(G)$, there is a u, v-path in G.

(3.9) A brute-force way of checking connectedness would be checking that u, v-path exists for every $u, v \in V(G)$, checking $\binom{|V(G)|}{2}$ paths in total. The following proposition proposes a better algorithm for doing this.

Proposition 3.4. *Let G be a graph and let $v \in V(G)$. If for every $u \in V(G)$ there exists a u, v-path in G, then G is connected.*

Proposition 3.4 means we only have to check $|V(G)| - 1$ paths to verify the connectedness of G.

Proof of Proposition 3.4. This result follows from Corollary 3.2.1 directly. ∎

(3.10) Disconnectedness
To show that a graph is disconnected, we need a pair of vertices where no path between them exists. But doing so is not straightforward, so wedesire to find another way to directly show disconnectedness: cuts.

Def'n 3.17. **Component, Cut** of a Graph
Let G be a graph.

(a) A ***component*** C of G is a maximal connected subgraph of G.

(b) Let $X \subseteq V(G)$. The ***cut*** induced by X in G is the set of all edges in G with exactly one end in X.

Proposition 3.5. *Let G be a graph. Then G is connected if and only if G has exactly 1 component.*

Theorem 3.6. Characterization of Disconnectedness
Let G be a graph. Then G is disconnected if and only if there exists a nonempty proper $X \subseteq V(G)$ such that the cut induced by X is empty.

Proof.

- ($\implies$) Since G is disconnected, G has at least 2 components, say H_1, H_2. Then $V(H_1)$ is a nonempty proper subset of $V(G)$. For the sake of contradiction, suppose that there exists $v \in V(H_1), u \in V(G) \setminus V(H_1)$ such that $uv \in E(G)$. Then the subgraph F with $V(F) = V(H_1) \cup \{v\}, E(F) = E(H_1) \cup \{uv\}$ is connected, contradicting the maximality of H_1. ◁
- ($\impliedby$) Suppose there exists a nonempty proper $X \subseteq V(G)$ such that the cut induced by X is empty. Since X is nonempty and proper, there exist vertices $v \in X, u \in V(G) \setminus X$. For the sake of contradiction, suppose that there exists a v, u-path, $(v_i)_{i=0}^{k}$ where $v_0 = v, v_k = u$. Since $v_0 \in X$, we can find the largest index i such that $v_i \in X$. Moreover, $v_k \notin X$, so $i < k$. So $i + 1 \leq k$, which means $v_i v_{i+1}$ is an edge in the cut induced by X, contradicting the fact that the cut is empty. ∎

Def'n 3.18. **Bridge** in a Graph

Let G be a graph. A ***bridge*** in G is an edge $e \in E(G)$ such that $G - e$ has more components than G.

Proposition 3.7.

Let G be a graph. If $e = uv \in E(G)$ is a bridge in G and H is the component such that $e \in E(H)$, then

(a) $H - e$ has two components; and

(b) u, v are in different components in $H - e$.

Proof.

(a) Suppose that $H - e$ has at least 3 components, for the sake of contradiction. This means there exists a component J of $H - e$ that does not have both u, v. Then $V(J)$ is a nonempty proper subset of $V(H)$, inducing an empty cut in $H - e$. Since $u, v \notin V(J)$, it follows that the cut induced by J in H is also empty. This means H is disconnected by Theorem 3.6, so we obtained the desired contradiction. ◁

(b) Suppose that there is a component J of $H - e$ such that $u, v \in V(J)$, for the sake of contradiction. We know that J is a proper component and that the cut induced by $V(J)$ in $H - e$ is empty by (a). But $u, v \in V(J)$, so $e = uv$ is not an element of the cut induced by $V(J)$ in H. This means the cut induced by $V(J)$ in H is empty, contradicting the connectedness of H. ∎

Corollary 3.7.1.

Consider the setting of Proposition 3.7. Then $G - e$ has one more component than G and u, v are in different components of $G - e$.

Theorem 3.8. Characterization of Bridges

Let G be a graph. An edge $e \in E(G)$ is a bridge of G if and only if e is not in any cycle of G.

Proof.

- ($\implies$) Suppose that $e = uv$ is in a cycle C, represented by $u, v, v_1, \ldots, v_k, u$. Then by removing uv from the cycle, we obtain a u,v-path $v, v_1, \ldots, v_k, u$. This path is also in $G - uv$, so u, v are in the same component of $G - uv$. Thus by Corollary 3.7.1, uv is not a bridge. ◁
- ($\impliedby$) Suppose that $e = uv$ is not a bridge. Suppose that H is the component of G containing E. Since uv is not a bridge, $H - uv$ is connected, so there exists a u,v-path $(v_i)_{i=0}^{k}$ in H, with $v_0 = u, v_k = v$. Then the subgraph C of H with $V(C) = \{v_0, \ldots, v_k\}$ and $E(C) = \{v_0v_1, \ldots, v_{k-1}v_k, v_kv_0\}$ is a cycle, and $uv = v_kv_0 \in E(C)$ by construction. ∎

Eulerian Circuits

Def'n 3.19. **Eulerian Cicuit** in a Graph

Let G be a graph. An ***eulerian circuit*** in G is a closed walk that uses every edge of the graph exactly once.

Theorem 3.9. Characterization of the Existence of Eulerian Circuit

Let G be a connected graph. Then G has an Eulerian circuit if and only if $\deg(v)$ is even for all $v \in V(G)$.

4.
Trees

4.1 Trees

Trees

Def'n 4.1.
Forest, Tree
A ***forest*** is an acyclic graph. A ***tree*** is a connected forest.

Proposition 4.1. *Every edge in a forest is a bridge.*

Proof. This result follows immediately from Theorem 3.8. ∎

Proposition 4.2. *Let T be a tree. Then $|E(T)| = |V(T)| - 1$.*

Proof. We proceed inductively.

- The result is clear when $V(T)$ is a singleton. ◁
- Suppose that for every tree with $1, \ldots, n-1$ vertices the result holds and let T be a tree with n vertices, where $n \in \mathbb{N}, n \geq 2$. Let $e \in E(T)$ (which exists by the inductive hypothesis). Then observe that $T - e$ has two components, say S, R, since every edge of a tree is a bridge. Moreover, clearly S, R are trees, so by the inductive hypothesis, $|E(S)| = |V(S)| - 1, |E(R)| = |V(R)| - 1$. Therefore,

$$|E(T)| = |E(S)| + |E(R)| + 1 = |V(S)| - 1 + |V(R)| - 1 + 1 = |V(T)| - 1,$$

as desired. ∎

Corollary 4.2.1. *Let F be a forest with k components. Then $|E(F)| = |V(T)| - k$.*

Def'n 4.2.
Leaf in a Tree
Let T be a tree. A ***leaf*** in T is a vertex $v \in V(T)$ with $\deg(v) = 1$.

Proposition 4.3. *Let T be a tree with $|V(T)| \geq 2$. Then T has at least 2 leaves.*

Proof. Let $(v_i)_{i=1}^k$ be a longest path in T. Since $|V(T)| \geq 2$, it follows that $k \geq 2$. Moreover, the only neighbor of v_1 is v_2, since otherwise we violate the maximality of $(v_i)_{i=1}^k$ or the acyclic property of T. Similarly v_{k-1} is the only neighbor of v_k. Thus we have v_1, v_k as leaves. ∎

(4.1) We can actually give an exact formula for determining the number of leaves in a tree. Then

$$|V(T)| = \sum\nolimits_{i \in \mathbb{N}} n_i,$$

where each n_i is the number of vertices in T of degree i. So by Proposition 4.2,

$$|E(T)| = -1 + \sum\nolimits_{i \in \mathbb{N}} n_i. \qquad [4.1]$$

Moreover, by the handshaking lemma,

$$2|E(T)| = \sum\nolimits_{i \in \mathbb{N}} i n_i. \qquad [4.2]$$

Hence by [4.1], [4.2],

Number of Leaves in a Tree

$$n_1 = 2 + \sum_{i \geq 3} (i-2) n_i. \tag{4.3}$$

Note that we have the constant term 2 in [4.3]: this corresponds to Proposition 4.3.

Proposition 4.4. *Let T be a tree. Then for every $u, v \in V(T)$, there exists a unique u,v-path in T.*

Proposition 4.5. *Let T be a tree. Then T is bipartite.*

Def'n 4.3. **Spanning Tree** of a Graph

Let G be a graph. A ***spanning tree*** of G is a spanning graph of G that is a tree.

Theorem 4.6. Characterization of Connectedness II

Let G be a graph. Then G is connected if and only if G has a spanning tree.

Proof.

- ($\Longrightarrow$) We proceed inductively.
  - If G is acyclic, then G is a spanning tree for itself.
  - Suppose that the result holds for every graph with $0, \ldots, k-1$ cycles, where $k \in \mathbb{N}$, and let G be a graph with k cycles. Choose any $e \in E(G)$ that is not a bridge and consider the graph $G-e$. Since the number of cycles in $G-e$ is strictly less than the number of cycles in G, by the inductive hypothesis, G has a spanning tree, say T. But note that T is also a spanning tree for G. ◁
- ($\Longleftarrow$) Suppose that G has a spanning tree, say T. Then given any $u, v \in V(G)$, note that there exists a u,v-path in T. But this path is also in G, so G is connected. ∎

Corollary 4.6.1. *Let G be a graph. If any of the following 3 conditions hold, then G is a tree.*

(a) G is connected.

(b) G is acyclic.

(c) G has $|V(G)|-1$ edges.

Proposition 4.7. *Let G be a graph and let T be a spanning tree of G.*

(a) For any $e \in E(G) \setminus E(T)$, $T+e$ has a unique cycle.

(b) For any $e \in E(G) \setminus E(T), f \in E(T)$, $T+e-f$ is a spanning tree of G.

Proof.

(a) Let $u, v \in V(G)$ be such that $uv = e$. Since there exists a unique u,v-path in T, it follows that the cycle in $T+e$ exists and is unique. ◁

(b) This follows immediately from Corollary 4.6.1. ∎

Proposition 4.8.

Let G be a graph and let T be a spanning tree of G.

(a) *For any $f \in E(T)$, $T - f$ has exactly 2 components.*

(b) *For any $f \in E(T)$, if e is an edge in the cut induced by the vertices of any component of $T - f$, then $T - f + e$ is a spanning tree of G.*

Theorem 4.9. Characterization of Bipartite Graphs

Let G be a graph. Then G is bipartite if and only if G has no odd cycle.

Proof.

- ($\Longrightarrow$) Suppose that G is bipartite and let $\{A, B\}$ be a bipartition for G. Let C be a cycle of length k, represented by the closed walk $v_1, \ldots, v_k, v_1$. Then $v_i v_{i+1}$ is an edge for all $i \in [k-1]$, so

$$v_i \in A \iff v_{i+1} \in B$$

for all $i \in [k-1]$. This means $v_1 \in A$ if and only if every v_i with odd $i \in [k]$ is in A. So if k is odd, then v_1, v_k are in the same set (i.e. A, B), which leads to a contradiction since $v_k v_1 \in E(G)$. Thus k is even. $\triangleleft$

- ($\Longleftarrow$) Suppose that G is not bipartite. Since we can find a component of G that is not bipartite, suppose that G is connected without loss of generality. This allows us to find a spanning tree T of G. Since any spanning tree is bipartite, let $\{A, B\}$ be a bipartition of T. Since G is not bipartite, there must exist an edge of G that joins 2 vertices, both from A or both from B. Without loss of generality, assume that $e = uv \in E(G)$ such that $u, v \in A$. Since T is connected, there exists a u,v-path $(v_i)_{i=0}^{k}$ with $v_0 = u, v_k = v$. But note that $(v_i)_{i=0}^{k}$ is of odd length, since T is bipartite and both *end*-vertices $v_0, v_k \in A$. Thus $C = \left(\{v_i\}_{i=0}^{k}, \{v_0 v_1, \ldots, v_{k-1} v_k, v_k v_0\}\right)$ is an odd cycle. ∎

Def'n 4.4.

Minimum Spanning Tree (MST) of a Graph

Let G be a connected graph and let $w : E(G) \to \mathbb{R}$, which is called a ***weight function*** on the set of edges. A spanning tree T of G whose sum

$$w(T) = \sum\nolimits_{e \in E(T)} w(e)$$

is minimum is called a ***minimum spanning tree*** (or ***MST*** for short) of G.

(4.2) Prim's Algorithm for MST

Suppose that we are given a graph G with a weight function $w : E(G) \to \mathbb{R}$, and consider the following algorithm.

(a) Choose any $v_i \in V(G)$ and let $T_1 = (\{v_i\}, \varnothing)$.

(b) For each $i \in \{2, \ldots, |V(G)|\}$, let $C \subseteq E(G)$ be the cut induced by T_{i-1}. Choose an edge $e_i = u_i v_i \in C$ with the smallest weight. Without loss of generality, assume $u_i \in V(T_{i-1}), v_i \notin V(T_{i-1})$. Define

$$T_i = (V(T_{i-1}) \cup \{v_i\}, E(T_{i-1}) \cup \{e_i\}).$$

We claim that this algorithm produces an MST.

Theorem 4.10. Correctness of Prim's Algorithm

Consider (4.2). $T_{|V(G)|}$ is an MST of G.

Proof. We proceed inductively, where we verify that, given any $i \in \mathbb{N}$, there exists a MST of G that contains T_i.

(a) $T_1 = (\{v_1\}, \varnothing)$, which is contained in every MST of G. ◁

(b) Let $i \in \{2, \ldots, |V(G)|\}$ and suppose that there exists an MST S of G containing T_{i-1} as a subgraph. We may assume that $e_i \notin E(S)$, since otherwise we are done. Consider $S + e_i$, which is not a tree and contains a unique cycle C. Now observe that there must exist an edge $f \in E(C)$ in the cut induced by T_{i-1} and $f \neq e_i$, since $S - e_i$ is a path joining a vertex in $V(T_{i-1})$ to a vertex in $V(G) \setminus V(T_{i-1})$. Note that by Proposition 4.8, $R = S + e_i - f$ is also a spanning tree of G. But since Prim's algorithm chooses an edge with the minimum weight, $w(e_i) \leq w(f)$, hence $w(e_i) = w(f)$ (since the other direction is clear from the fact that $f \in E(S)$, where S is an MST). Therefore R is also an MST of G, and T_i is contained in R. ∎

This page intentionally left blank.

5. Planar and Nonplanar Graphs

5.1 Planar Graphs
5.2 Euler's Formula and Platonic Solids
5.3 Nonplanar Graphs
5.4 Coloring
5.5 Planar Dual

Planar Graphs

Def'n 5.1. **Planar Embedding** of a Graph
Let G be a graph. A ***planar embedding*** of a graph G is a drawing of the graph on the plane such that vertices are at different points, and edges intersect only at their common endpoints (i.e. edges do not cross). A graph that has a planar embedding is a ***planar*** graph.

Def'n 5.2. **Face** of a Planar Embedding
A ***face*** of a planar embedding is a connected region on the plane (not separated by the edges).

Def'n 5.3. **Boundary** of a Face
The ***boundeary*** of a face is a subgraph of all vertices and edges that touch the face. Two faces are ***adjacent*** if they share at least one edge in their boundaries.

Def'n 5.4. **Boundary Walk, Degree** of a Face
For a connected planar embedding, the ***boundary walk*** of a face is a closed walk once around the perimeter of the face boundary. The ***degree*** of a face f is the length of its boundary walk, denoted as $\deg(f)$.

(5.1) In case the graph is not connected, we sum up the length of connected components.

Theorem 5.1. Handshaking Lemma for Faces

Let G be a planar graph and let F be the set of all faces. Then

$$\sum_{f\in F}\deg(f)=2\left|E(G)\right|.$$

Proof. Eacy edge contributes 2 to the sum of face degrees, one for each side of the edge. ∎

Proposition 5.2.

Let G be a planar graph. $e\in E(G)$ is a bridge if and only if the two sides of e are in the same face.

(5.2) To show Proposition 5.2, we require a topological fact called *Jordan curve theorem*, which we are going to rephrase as follows.

Theorem 5.3. Jordan Curve Theorem for Planar Embeddings

Every planar embedding of a cycle separates the plane into two parts, one on the inside, one on the outside.

Euler's Formula and Platonic Solids

Theorem 5.4.
Euler's Formula

Let G be a connected planar graph with V vertices and E edges. Consider a planar embedding of G with F faces. Then

$$V - E + F = 2. \tag{5.1}$$

Def'n 5.5. **Platonic** Connected Planar Graph

Let G be a connected planar graph. We say G is ***Platonic*** if it has a planar embedding where every vertex has the same degree greater than or equal to 3 and every face has the same degree greater than or equal to 3.

(5.3) Suppose a Platonic graph G has vertex degree $d_v \geq 3$ and face degree $d_f \geq 3$. Then we obtain the following equations.

- *handshaking lemma*: $V = \frac{2E}{d_v}$.
- *handshaking lemma for faces*: $F = \frac{2E}{d_f}$.
- *Euler's formula*: $V - E + F = 2$.

By substituting the results of handshaking lemmas into Euler's formula, we obtain

$$\frac{2E}{d_v} - E + \frac{2E}{d_f} = 2.$$

After rearranging,

$$E\left(2d_f - d_v d_f + 2d_v\right) = 2d_v d_f.$$

This means $2d_f - d_v d_f + 2d_v > 0$, so

$$\left(d_v - 2\right)\left(d_f - 2\right) < 4. \tag{5.2}$$

[5.2] gives every possible (d_v, d_f) pairs: $(3,3)$,[1] $(3,4)$,[2] $(3,5)$,[3] $(4,3)$,[4] $(5,3)$.[5]

Nonplanar Graphs

(5.4) Unlike showing that a graph is planar, a drawing is not sufficient, since we have to show that every drawing has *crossing edges*. One way to prove nonplanarity is to show that there are *too many edges*. In other words, there is maximum number of edges that a planar graph (with certain properties) can have. If a graph has more edges, then it must be nonplanar.

[1] $(3,3)$ is called a *tetrahedron*.
[2] $(3,4)$ is called a *cube*.
[3] $(3,5)$ is called a *dodecahedron*.
[4] $(4,3)$ is called *octahedron*.
[5] $(5,3)$ is called a *icosahedron*.

Proposition 5.5. *Let G be a planar graph with n vertices an m edges. If there is a planar embedding of G where every face has degree at least $d \geq 3$, then $m \leq \frac{d(n-2)}{d-2}$.*

Proof. We may assume that G is connected. Let F be the set of all faces in the embedding, with $|F| = s$. By the handshaking lemma for faces,

$$2m = \sum_{f \in F} \deg(f) \geq \sum_{f \in F} d = d_s. \tag{5.3}$$

By Euler's formula,

$$s = 2 - n + m. \tag{5.4}$$

Combining [5.3], [5.4] gives

$$2m \geq d_s = d(2 - n + m),$$

and by rearranging,

$$m \leq \frac{d(n-2)}{d-2}.$$

∎

Proposition 5.6. *Let G be a planar graph. If G contains a cycle, then in any planar embedding of G, every face boundary contains a cycle.*

Theorem 5.7. *Let G be a planar graph with $n \geq 3$ vertices and m edges. Them $m \leq 3n - 6$.*

Proof. If G does not have a cycle, then G is a forest, so $m \leq n - 1 \leq 3n - 6$ whenever $n \geq 3$. So we may assume that G has a cycle. This means every face boundary has a cycle by Proposition 5.6. Since each cycle has length at least 3, every face has degree at least 3. So by Proposition 5.5, $m \leq \frac{3(n-2)}{3-2} = 3n - 6$. ∎

Corollary 5.7.1. *K_5 is nonplanar.*

Proof. Note that K_5 has 5 vertices and 10 edges. By Theorem 5.7, any planar graph with 5 vertices has at most $3 \cdot 5 - 6 = 9$ edges. So K_5 is nonplanar. ∎

Corollary 5.7.2. *Consider the setting of Theorem 5.7. If G is bipartite, then $m \leq 2n - 4$.*

Proof. A proof is essentially the same as the presented proof of Theorem 5.7, where we note that every cycle has lengt at least 4 by the fact that G is bipartite, so every face has degree at least 4. ∎

Corollary 5.7.3. *$K_{3,3}$ is nonplanar.*

Proof. $K_{3,3}$ is bipartite and has 6 vertices, and by Corollary 5.7.2, a planar bipartite graph with 6 vertices can have at most 8 edges. But $K_{3,3}$ has 9 edges, so is nonplanar. ∎

(5.5) We showed that $K_5, K_{3,3}$ are nonplanar, and it immediately follows that any graph containing K_5 or $K_{3,3}$ is nonplanar. We now claim that any graph that *looks like* $K_5, K_{3,3}$ is nonplanar. To be more precise, we mean any graph obtained by *replacing each edge in* $K_5, K_{3,3}$ *by a path*.

Def'n 5.6. **Edge Subdivision** of a Graph
Let G be a graph. An ***edge subdivision*** of G is a graph obtained by replacing each edge of G with a new path of length at least 1.

Theorem 5.8.
Kuratoski's Theorem

Let G be a graph. Then G is planar if and only if it does not contain an edge subdivision of K_5 or $K_{3,3}$ as a subgraph.

Coloring

Def'n 5.7. **k-coloring**
Let G be a graph and let C be a set of size k. A ***k-coloring*** of a graph G is a function $f : V(G) \to C$ such that $f(u) \neq f(v)$ for all adjancet $u, v \in V(G)$. If such function exists, then we say G is ***k-colorable***.

(5.6)
Coloring

We have two extreme cases of coloring.

Proposition 5.9. *A graph G is 2-colorable if and only if G is bipartite.*

Proposition 5.10. *For any graph G, G can be colored using $|V(G)|$ colors.*

Corollary 5.10.1. *Let $n \in \mathbb{N}$. Then K_n is n-colorable but not k-colorable for all $k < n$.*

We will focus on coloring planar graphs.

Theorem 5.11.
6-color Theorem

Every planar graph is 6-colorable.

Lemma 5.11.1. *Every planar graph has a vertex of degree at most 5.*

Proof. Let G be a planar graph with $n \in \mathbb{N}$ vertices and suppose, for the sake of contradiction, that every vertex has degree at least 6. By the handshaking lemma, the number of edges in G is at least $3n$, which contradicts Theorem 5.7. ∎

Proof of Theorem 5.11.. We proceed inductively.

- Any graph with 1 vertex is 6-colorable. ◁
- Suppose that the result holds for every graph with $n-1$ vertices, where $n \in \mathbb{N}, n \geq 2$. Let G be a graph with n vertices and let $v \in V(G)$ be a vertex of degree at most 5, which exists by Lemma 5.11.1. Now consider the graph H obtained by removing v and every edges incident to v in G. By the inductive hypothesis, H has a 6-coloring. Now consider the neighbors of v in G: since $\deg(v) \leq 5$, we have at least one color that is not used by neighbors of v. Thus G is 6-colorable.

In fact, we can prove the following stronger result.

Theorem 5.12.
5-color Theorem

Every planar graph is 5-colorable.

To prove Theorem 5.12, we introduce the following notion of *edge contraction*.

Def'n 5.8. **Edge Contraction** in a Graph
Let G be a graph and let $u, v \in V(G)$ be neighbors. The ***contraction*** of uv results in a new graph $G/uv = (V', E')$, where $V' = V(G) \setminus \{u, v\} \cup \{w\}$, and $x, y \in V'$ are adjacent if and only if $xy \in E(G)$ (in case $x, y \neq w$) or, if $x = w$, then y is a neighbor of u or v in G (and vice versa). We call G/uv an ***edge contraction*** of G.

The following result follows immediately from Def'n 5.8.

Lemma 5.12.1. *If G is planar and let $e \in E(G)$. Then G/e is planar.*

Proof of Theorem 5.12. We proceed inductively.

- Clearly any graph with 1 or 2 vertices is 5-colorable.
- Assume that the result holds for every graph with less than n vertices, where $n \in \mathbb{N}, n \geq 3$. Let G be a planar graph with n vertices. Let $v \in V(G)$ be such that $\deg(v) \leq 5$, which exists by Lemma 5.11.1. If $\deg(v) \leq 4$, then by using the same approach as we did in the proof of Theorem 5.11, we are done. So assume that $\deg(v) = 5$. Now observe that, if every neighbors of v are adjacent, then the neighbors of v form K_5, which contradicts the planarity of G. So fix $u, w \in V(G)$, neighbors of v that are not adjacent. Let G' be obtained from G by contracting vu, vw. Then by Lemma 5.12.1, G' is planar. Let x be the contracted vertex. Then G' is planar with $n-2$ vertices, so 5-colorable by the inductive hypothesis. Now, suppose that $f : V(G') \to C$ is a 5-coloring of G', where C is a set of cardinality 5. We obtain a 5-coloring $\tilde{f} : V(G) \to C$ as follows. Define
$$\tilde{f}(y) = \begin{cases} f(x) & \text{if } y = u \text{ or } y = w \\ f(y) & \text{if } y \notin \{v, u, w\} \end{cases}$$
for all $y \in V(G) \setminus \{v\}$. Note that we are using only 4 colors for the neighbors of v, so we may define $\tilde{f}(v)$ to be the remaining color. Observe that $\tilde{f}$ is a coloring since u, w are not adjacent. ∎

The strongest form is the following.

Theorem 5.13.
4-color Theorem

Every planar graph is 4-colorable.

We only have computer-aided proofs for 4-color theorem. Also note that K_4 is planar but not 3-colorable, so indeed Theorem 5.13 is the strongest k-colorability theorem that holds for every planar graph.

Planar Dual

Def'n 5.9. **Dual Graph** of a Planar Graph
Let G be a planar graph. The ***dual graph*** G^* of G has a vertex v_f for each face f in the embedding of G and an edge $v_{f_1}v_{f_2}$ for each edge e in G with faces f_1, f_2 on the two sides.

Note that, even when G is a planar graph, we may have G^* that has loops and multiple edges.

Proposition 5.14.

Let G be a planar graph. Then G^* is planar.

Now that we have the notion of dual graphs, we can translate the notion of *coloring vertices* to *coloring faces*.

Theorem 5.15. 4-color Theorem for Faces

Every planar embedding is 4-(face-)colorable.

(5.7) Properties of the Dual

Here are some interesting facts about the dual. Let G be a planar graph.

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

(b) A vertex in G uniquely corresponds to a face in G^* of the same degree.

(c) If G is platonic, then G^* is also platonic.

(d) If we draw a closed curve on 2-dimensional plane, then the drawing is 2-(face-)colorable.

This page intentionally left blank.

6.
Matchings

6.1 Matchings
6.2 Konig's Theorem and Hall's Theorem

Matchings

Def'n 6.1. **Matching** of a Graph
A ***matching*** of a graph is a set of edges where no two edges share a common vertex.

(6.1) Note that, given a graph, it is easier to find a small matching. In fact, $\varnothing$ is a matching of any graph. So in general we are interested in finding a matching with maximum size.

Def'n 6.2. **Perfect** Matching
Let G be a graph and let M be a matching of G. We say $v \in V(G)$ is ***saturated*** if v is incident with an edge in M. If every $v \in V(G)$ is saturated, we say M is ***perfect***.

Def'n 6.3. **Cover** of a Graph
A ***cover*** C of a graph G is a subset of $V(G)$ such that each edge in G has at least one endpoint in C.

(6.2) For covers, it is easier to find larger cover given a graph ($V(G)$ is a cover of any graph G). Hence we are interested in finding a cover with minimum size.

Proposition 6.1. *Let G be a graph. Then for any matching M and cover C of G,*

$$|M| \leq |C|.$$

Proof. For every $uv \in M$, observe that at least one of u, v is in C since C is a cover. Since no two edges in M share a vertex, C must include at least $|M|$ distinct vertices, one from each edge of M. ∎

Proposition 6.2. *Let G be a graph. If M is a matching and C is a cover of G with $|M| = |C|$, then M is a maximum matching of G and C is a minimum cover of G.*

Konig's Theorem and Hall's Theorem

Theorem 6.3.
Konig's Theorem
Let G be bipartite. Then the size of a maximum matching is equal to the size of a minimum cover.

(6.3) To prove Konig's theorem, we first develop necessary tools. Now, given a graph G and a matching M of G, how do we find a matching of size larger than M? If there exists an edge of G whose endpoints are not saturated by M, then we can certainly include this edge to make a larger matching. However, this is not always possible, and when there is no such edge, we attempt to use the following notion.

Def'n 6.4. **Augmenting Path** with respect to a Matching
Let G be a graph and let M be a matching of G. An ***alternating path*** P with respect to M is a path where consecitive edges alternate being in M and not in M. We say an alternating path Q is an ***augmenting path*** if it starts and ends in distinct unsaturated vertices.

Proposition 6.4. *Let G be a graph and let M be a matching. If M has an augmenting path, then M is not maximum.*

Proof. Let P be an augmenting path with respect to M. Then observe that

$$N = M \setminus (E(P) \cap M) \cup (E(P) \setminus M)$$

has 1 more edge than M. ∎

(6.4) Since we are going to prove Konig's theorem by introducing an algorithm that looks for augmenting path, let us see how an augmenting path looks like in a bipartite graph. Let G be a bipartite graph with bipartition $\{A, B\}$. Without loss of generality, suppose that an augmenting path starts at A. Then, besides the starting vertex, every time we reach a vertex in A in the path, it is through a matching edge, so the vertex is saturated. This means every augmenting path starts and ends with different parts of the bipartition. This means we can start with all unsaturated vertices in A, fan out to find an unsaturated vertex in B.

(6.5) Bipartite Matching Algorithm

Let G be bipartite with bipartition $\{A, B\}$. We desire to find a maximum matching in G. The idea is to find all possible alternating paths starting from unsaturated vertices in A. If an augmenting path is found, then we update the matching to get a larger one by Proposition 6.4. Otherwise, we find a cover whose size is equal to that of the matching. This proves that the matching is maximum.

(a) Find a matching M of G.

(b) Let X be the set of all unsaturated vertices in A and $Y = \varnothing$.

(c) Find all neighbors of vertices in X in $B \setminus Y$.

  (i) If one such vertex is unsaturated, then we have found an augmenting path. By *swapping edges*, we obtain a matching of size one larger than M. Go to step (b).

  (ii) If all such vertices are saturated, then put all of them in Y. Add their matching neighbors to X. Go to step (c).

  (iii) If no such vertices exist, then the algorithm halts. The matching is maximum with minimum cover $Y \cup (A \setminus X)$.

Proof of Konig's Theorem. Let M be the matching obtained at the end of the bipartite matching algorithm and let X_0 be X before iterations. We have the following claims.

(a) There is no edge between $X, B \setminus Y$.

Proof. If there exists such edge, then we can use this edge to extend an alternating path, so the algorithm would include this edge in Y, so a contradiction. ◁

(b) Every vertex in Y is saturated.

Proof. If there is an unsaturated vertex in Y, then we have an augmenting path, so the algorithm would not terminate. ◁

(c) Every vertex in $A \setminus X$ is saturated.

Proof. This is because, in the matching M_0 that we started the algorithm, all unsaturated vertices of A are in $X_0 \subseteq X$. ◁

(a) means every edge has at least one ed in $A \setminus X$ or Y, implying that $Y \cup (A \setminus X)$ is a cover. Moreover, observe that every matching edge that saturates a vertex in Y, the other end is in X by the algorithm This also means if a matching edge that saturates a vertex in $A \setminus X$, the other end is in $B \setminus Y$. This means the matching edges saturating $Y, A \setminus X$ are distinct, so

$$|M| = |Y| + |A \setminus X|,$$

which is equal to the size of the cover. ∎

Corollary 6.4.1. *Let G be bipartite with m edges and* $d = \max_{v \in V(G)} (\deg(v))$. *Then G has a matching of size at least* $\frac{m}{d}$.

Proof. Observe that every vertex covers at most d edges, so we nned at least $\frac{m}{d}$ vertices to cover all m edges. The rest follows from Konig's theorem. ∎

Def'n 6.5. **Neighbor Set** of a Vertex Set

Let G be a graph and let $D \subseteq V(G)$. The ***neighbor set*** of D, denoted as $N_G(D)$ or $N(D)$ when G is understood, is the set of all vertices adjacent to at least one vertex in D.

Theorem 6.5. Hall's Theorem

A bipartite graph G with bipartition $\{A, B\}$ *has a matching that saturates all vertices in A if and only if for all* $D \subseteq A$, $|N(D)| \geq |D|$.

Proof.

- ($\implies$) Suppose M is a matching that saturates A. Then for any $D \subseteq A$, each matching edge that saturates a vertex in D has the other end in $N(D)$. Since no matching edges share vertices, these matching neighbors are distinct. So $|N(D)| \geq |D|$. ◁

- Suppose G does not have a matching that saturates A. We desire to find a subset $D \subseteq A$ which violates *Hall's condition* (i.e. $|N(D)| < |D|$). Let M be a maximum matching in G. Then $|M| < |A|$ since M does not saturate every vertex in A. By Konig's theorem, there exists a cover C of G with $|C| = |M|$, which also implies $|C| < |A|$. Now consider the following 4 sets:

$$A \cap C, A \setminus C, B \cap C, B \setminus C.$$

Since C is a cover, no edge joins $A \setminus C$ with $B \setminus C$. So all neighbors of $A \setminus C$ in particular are in $B \cap C$, which means

$$N(A \setminus C) \subseteq B \cap C. \tag{6.1}$$

Now

$$C = (C \cap A) \cup (C \cap B), A = (A \cap C) \cup (A \setminus C).$$

Also, $|C| < |A|$. So it follows that

$$|C \cap B| < |A \setminus C|. \tag{6.2}$$

But [6.1] implies that $|N(A \setminus C)| \leq |B \cap C|$, so

$$|N(A \setminus C)| \overset{[6.1]}{\underset{}{\overset{[6.2]}{<}}} |A \setminus C|,$$

so $A \setminus C$ is a subset of A violating Hall's condition. ∎

Corollary 6.5.1. *If G is k-regular bipartite with $k \in \mathbb{N}$, then G has a perfect matching.*

Proof. Observe that any bipartition $\{A, B\}$ of G satisfies $|A| = |B|$. This means any matching that saturates A is a perfect matching. Moreover, given any $D \subseteq A$, each edge with one edn in D has the other end in $N(D)$, which means

$$\sum\nolimits_{v \in D} \deg(v) \leq \sum\nolimits_{v \in N(D)} \deg(v). \tag{6.3}$$

Simce G is k-regular, [6.3] is equivalent to

$$k|D| \leq k|N(D)|.$$

Since $k \geq 1$, it follows that $|D| \leq |N(D)|$, so Hall's condition is met. This means a matching that saturates A exists, which is a perfect matching as we noted. ∎

Corollary 6.5.2. *The edges of a k-regular bipartite graph can be partitioned into k perfect matchings.*

Proof. Use induction on k. ∎