Compact course notes

Math 249, Fall 2010

Introduction to Combinatorics

Professor: D. Wagner
transcribed by: J. Lazovskis
University of Waterloo
December 16, 2010

Contents

File I
Enumeration

1 Relations and strings

1.1 Recurrence relations

The Fibonacci numbers f_n are given by: $f_n = \frac{1+\sqrt{5}}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{\sqrt{5}-1}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$

Theorem 1.1.1. [The Binomial Theorem]
For any rational number a, $(1+x)^a = \sum_{k\geqslant 0}\binom{a}{k}x^k$

1.2 Binary strings

Definition 1.2.1. A binary string is $\sigma = b_1b_2\dots b_n$ such that each $b_i \in \{0,1\}$ for $1 \leqslant i \leqslant n$.

Definition 1.2.2. The length of $\sigma = b_1b_2\dots b_n$ is given by $\ell(\sigma) = n$.

The infinite set of all binary strings is given by $\{0,1\}^*$.
The unique binary string of length 0 is denoted by ε and is denoted the empty string.

Definition 1.2.3. Given a set of binary strings A, the generating function of A is defined to be

$$\Phi_A(x) = \sum_{\sigma\in A} x_1^{\omega_1(\sigma)}x_2^{\omega_2(\sigma)}\cdots x_m^{\omega_m(\sigma)} = \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}\cdots\sum_{n_m=0}^{\infty} c_{n_1}x_1^{n_1}c_{n_2}x_2^{n_2}\cdots c_{n_m}x_m^{n_m}$$

where $[x_1^n] = c_{n_i} = \omega_i(\sigma)$, where ω_i is a function that keeps track of a certain property of σ.

1.3 Regular languages

Definition 1.3.1. A regular language is

Proposition 1.3.2. [Properties of regular languages
· $\{\varepsilon\}, \{0\}, \{1\}, \emptyset$ are regular languages
· If A and B are regular languages, then their union $A \cup B$ is a regular language
· If A and B are regular languages, then their concatenation $AB = \{\alpha\beta \,\big|\, \alpha\in A, \beta\in B\}$ is also
· If A is a regular language, then its iteration $A^* = \{\alpha_1\alpha_2\dots\alpha_n \,\big|\, n\in\mathbb{N} \text{ and each } \alpha_i \in A\}$ is also

Proposition 1.3.3. Every finite set $A \subset \{0,1\}^*$ is a regular language.

Definition 1.3.4. Given sets of binary strings A, B, their union is said to be ambiguous if their intersection is non-empty. Similarly, their concatenation is said to be ambiguous when the same string may be constructed in more than one unique way.

Eg. $\{011, 101\} \cup \{10100, 101\}$ is ambiguous
$\{011, 01\}\{10, 0\}$ is ambiguous

Proposition 1.3.5. [Construction of generating functions]
· If $A\cup B$ is unambiguous, then $\Phi_{A\cup B}(x) = \Phi_A(x) + \Phi_B(x)$
· If AB is unambiguous, then $\Phi_{AB}(x) = \Phi_A(x)\Phi_B(x)$

1.4 Block decompositions

Definition 1.4.1. Let σ be a string. Then a block is a maximal substring of consecutive equal bits.

Remark 1.4.2. The following are common block decompositions:

$$\begin{aligned}\{0,1\}^* &= 0^*(1^*1\ \ 0^*0)^*1^* \\ &= 1^*(0^*0\ \ 1^*1)^*0^* \\ \{0,1,2\}^* &= \{0,1\}^*(2^*2(\{0,1\}^* \setminus \{\varepsilon\}))^*2^* \\ &= \{1,2\}^*(0^*0(\{1,2\}^* \setminus \{\varepsilon\}))^*0^* \\ &= \{2,0\}^*(1^*1(\{2,0\}^* \setminus \{\varepsilon\}))^*1^*\end{aligned}$$

Theorem 1.4.3. Let $\mathcal{D} \subset \{1,\dots,b\}^*$ be the set of strings with no two consecutive equal bits. Then the generating function for $\mathcal{D}$ is

$$D(x_1,\dots,x_b) = \frac{1}{1-\left(\frac{x_1}{1+x_1}+\cdots+\frac{x_b}{1+x_b}\right)} = \left(1-\sum_{i=1}^{b}\frac{x_i}{1+x_i}\right)^{-1}$$

where x_i is the generating function for i^*i, with respect to the b-ary alphabet it came from.

Remark 1.4.4. The set $\{1,2,\dots,b\}^*$ has generating function $\frac{1}{1-bx}$.
The number of strings of length n in this set is b^n.

Theorem 1.4.5. If α is a self-avoiding b-ary string such that $\mathcal{N} \subset \{1,2,\dots,b\}$ does not contain α as a substring, then $\{1,2,\dots,b\} = \mathcal{N}(\alpha\mathcal{N})^*$. Further, if $|\alpha| = a$, then $\Phi_{\mathcal{N}}(x) = \frac{1}{1-bx+x^a}$.

1.5 Multisets

The expression $\binom{m+k-1}{m-1}$ is the number of $(m-1)$-element subsets of a set of size $m+k-1$ $\forall k \in \mathbb{N}$.

The coefficient of x^k in $\sum_{k=0}^{\infty}\binom{m+k-1}{m-1}x^k = \frac{1}{(1-x)^m} = \sum_{\{n_1,\dots,n_m\}\in\mathbb{N}} x^{n_1+\cdots+n_m}$

is the number of sequences $\{n_1,n_2,\dots,n_m\}$ with each $n_i \in \mathbb{N}$ and $n_1+n_2+\cdots+n_m = k$.

Remark 1.5.1. There is an isomorphism between

$$\Big(\text{ multisets of size } n \text{ with } t \text{ types }\Big) \text{ and } \Big((t-1)\text{-element subsets of } \{1,2,\dots,n+t-1\}\Big)$$

Proposition 1.5.2.
The probability of having exactly k p-type elements is a set of length n containing t types of elements is

$$P = \frac{{}_{n+t-1-k}C_{t-1}}{\binom{n+t-1}{t-1}}$$

2 Paths and trees

2.1 Latice paths

· On a square lattice grid, the number of ways to get from $(0,0)$ to (a,b) by only moving N (north) and E (east) is $\binom{a+b}{a} = \binom{a+b}{b}$. Given the restriction that the path to (a,b) may not cross the diagonal $x=y$, the number of paths is $c(a,b) = \left(1-\frac{a}{b+1}\right)\binom{a+b}{b}$

Definition 2.1.1. With respect to above, if (a, b) lies on the diagonal $x = y$, then $c(n, n) = \frac{1}{n+1}\binom{2n}{n}$ is the n-th Catalan number.

Moreover, these paths are then termed super-diagonal lattice paths, or Dyck paths.

Remark 2.1.2. There exists a bijection between Dyck paths and well-formed parenthesizations.

2.2 Plane planted trees

Definition 2.2.1. A plane planted tree has

· a root node
· a finite number of nodes
· every node has $k \geqslant 0$ children

Theorem 2.2.2. [Lagrange implicit function theorem]
Let $G(u)$ be a power series with non-zero constant term, i.e $[x^0]G(u) \neq 0$. Then

1. There exists a unique power series $R(x)$ such that $R(x) = xG(R(x))$
2. $[x^0]R(x) = 0$ and for all $n \geqslant 1$, $[x^n]R(x) = \frac{1}{n}[u^{n-1}]G(u)^n$.

Remark 2.2.3. There is a bijection among every pair of

· plane-planted trees
· well-formed parenthesizations
· binary root rees

File II
Graph theory

3 Walks, trails and paths

3.1 Base definitions

Definition 3.1.1. A graph, denoted by $G = (V, E)$ where

· V is a finite set of vertices
· E is a finite set of 2-element subsets of V, termed edges

Definition 3.1.2. The degree of a vertex v in a graph $G = (V, E)$ is the number of edges connected to v, or the number of occurrences of v in E.

Definition 3.1.3. A graph is said to be k-regular is every vertex of the graph has degree k.

Remark 3.1.4. More general graphs are termed multigraphs or directed graphs. A subset of these, simple graphs, are discussed below, and do not contain directed edges, multiples edges, or loops.

Definition 3.1.5. Graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic if there is a function $f : V \to W$ s.t.

· f is a bijection
· For any $v, w \in W$, $\{f(u), f(w)\} \in F \iff \{v, w\} \in E$

This relationship is then denoted by $G \cong H$.

Definition 3.1.6. For a graph $G = (V, E)$, a subgraph of G is a graph $H = (W, F)$ with $W \subset V$ and $F \subset E$.

Definition 3.1.7. If a subgraph $H = (W, F)$ of $G = (V, E)$ such that F consists of edges in G with both ends in W, then H is termed an induced subgraph of G.

Note that $(\emptyset, \emptyset)$ is a subgraph of every graph.

Definition 3.1.8. A perfect matching of a graph is a 1-regular spanning subgraph.

Definition 3.1.9. A walk in a graph $G = (V, E)$ is a sequence $W = v_0e_1v_1 \dots e_kv_k$ for $v_i \in V$ and $e_i = \{v_{i-1}, v_i\} \in E$. Vertices and edges in a walk do not have to be distinct.

Definition 3.1.10. A trail is a walk with no repeated edges.

Definition 3.1.11. A path is a trail with no repeated vertices.

3.2 Networking and connectedness

Definition 3.2.1. For a graph $G = (V, E)$, define a relation R on V by putting xRy for $x, y \in V$ if there exists a walk $W : x = v_0e_1v_1 \dots e_kv_k = y$ from x to y in G. Then we say that x reaches y.

Definition 3.2.2. The equivalence classes of R induce subgraphs of G termed the (connected) components of G. Further, G is connected if and only if it contains one component.

Definition 3.2.3. A cycle is a 2-regular connected graph.

Proposition 3.2.4. Let $G = (V, E)$ be a connected graph with $x, y \in V$. If there is a walk in G from x to y, then there is a path in G from x to y.

Definition 3.2.5. The distance from x to y in G is the minimum number of edges of any path from x to y in G, if there exists such a path. If x and y are in different components, then their distance between them is defined to be ∞.

Definition 3.2.6. A subgraph H of G is spanning if H contains all the vertices of G. Note that G is the only spanning-induced subgraph of G.

Definition 3.2.7. A Hamilton cycle in a graph G is a spanning subgraph that is a cycle.

Definition 3.2.8. A Gray code is a Hamilton cycle in Q_d, where Q_d denotes the d-dimensional cube.

Theorem 3.2.9. For all $d \geqslant 2$, Q_d has a Hamilton cycle.

3.3 Minimally connected graphs

Definition 3.3.1. If $G = (V, E)$ is a connected graph, then an edge $e \in E$ is a cut edge if $G \setminus e$ is not connected.

Definition 3.3.2. A minimally connected graph is such that every edge of the graph is a cut edge.

Proposition 3.3.3. Let $G = (V, E)$ be connected. Then $e \in E$ is a cut edge if and only if e is not contained in any cycle of G.

Lemma 3.3.4. Let $G = (V, E)$ be a connected graph, and let $e \in E$ be a cut edge $e = \{x, y\}$. Then $G \setminus e$ has exactly two components X, Y with $x \in V(X)$ and $y \in V(Y)$.

Theorem 3.3.5. A graph $G = (V, E)$ is minimally connected if and only if G is connected and contains no cycles.

Definition 3.3.6. A connected graph G which contains no cycles is a tree. A graph which only contains trees is a forest.

Proposition 3.3.7. If T is a tree with $p \geqslant 2$ vertices, then T has at least 2 vertices of degree 1.

Proposition 3.3.8. A tree with p vertices has $p - 1$ edges.

Theorem 3.3.9. [2-OUT-OF-3 THEOREM]
Let $G = (V, E)$ be a graph with p vertices and q edges. Any two of the three conditions below together imply the third:

1. G is connected
2. G contains no cycles
3. $q = p - 1$

Proposition 3.3.10. Every connected graph contains a spanning tree, and hence if it has p vertices, then it has at least $p - 1$ edges.

Lemma 3.3.11. [HANDSHAKE LEMMA]
Let $G = (V, E)$ be a graph with q edges. Then $\sum_{v \in V} \deg(v) = 2q$.

4 Planar graphs

4.1 Definitions

Definition 4.1.1. A plane embedding of a graph $G = (V, E)$ is a set $\{p(v) \mid v \in V\}$ of distinct points in $\mathbb{R}^2$ indexed by V and distinct curves $\{\gamma_e \mid e \in E\} \subset \mathbb{R}^2$ indexed by E such that

1. If $e = \{x, y\}$, then γ_e has endpoints $p(x), p(y)$
2. Each γ_e is a simple curve
3. Each γ_e does not contain $p(v)$ unless $v \in e$
4. If γ_e and γ_f intersect, then they intersect only at a common endpoint

Definition 4.1.2. A graph is planar if it has a plane embedding. Note that a graph may have more than 1 unique plane embedding.

Definition 4.1.3. A subdivision of a graph is the discussed graph with vertices of degree 2 added on the edges of the graph.

4.2 Kuratowski and Euler

Definition 4.2.1. A complete graph K_n is an $(n-1)$-regular graph with n vertices. That is, each vertex is connected to every other vertex by an edge.

Note that K_i for $i \in [1, 4]$ is planar, whereas K_j for $j \geqslant 5$ fail to be.

Definition 4.2.2. A complete bipartite graph $K_{a,b}$ is a graph $G = (V, E)$ such that $V = A \cup B$ and $A \cap B = \emptyset$, with $|A| = a$ and $|B| = b$, and every vertex in A is connected to every vertex in B.

Note that $K_{3,3}$ is not planar.

Theorem 4.2.3. [KURATOWSKI'S THEOREM]
A graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$ as a subgraph.

Definition 4.2.4. The connected components of $\mathbb{R}^2 \setminus \left(\bigcup_{e \in E} \gamma_e \right)$ are termed the faces of the embedded graph.

Definition 4.2.5. The degree of a face F is the sum of the lengths of the closed walks around the boundary of F.

Proposition 4.2.6. [HANDSHAKE LEMMA FOR FACES]
For any graph $G = (V, E)$ properly embedded in the plane, $\sum_{\text{faces } F} deg(F) = 2|E|$

Lemma 4.2.7. Let $G = (V, E)$ be a graph properly embedded in the plane. Let $e \in E$. Let F and F' be the faces on the two sides of the curve γ_e. Then $F = F' \iff e$ is a cut edge of E.

Theorem 4.2.8. [Euler's formula]
Let $G = (V, E)$ be a graph properly embedded in the plane. Then if G has

$$\left.\begin{array}{l} p \text{ vertices} \\ q \text{ edges} \\ r \text{ faces} \\ c \text{ connected components} \end{array}\right\} \text{then } p - q + r = c + 1$$

Proposition 4.2.9. Let $G = (V, E)$ be a graph properly embedded in the plane with p vertices and $q \geqslant 2$ edges. Then $q \leqslant 3p - 6$.

Proposition 4.2.10. Let $G = (V, E)$ be a graph properly embedded in the plane with p vertices and q edges and no cycles of length 3. Then $q \leqslant 2p - 4$.

Definition 4.2.11. The girth of a graph $G = (V, E)$ is the minimum length of a cycle in G (or defined to be ∞ if G is a tree).

Definition 4.2.12. Let $G = (V, E)$ be a connected graph. For each $v \in V$, let $c(v)$ be the maximum value of $\operatorname{dist}_G(v, w)$ for all $w \in V$. The radius of G is defined to be the maximum value of $c(v)$ for all $v \in V$.

4.3 Plane coloring

Definition 4.3.1. A proper coloring of a graph $G = (V, E)$ properly embedded in the plane is a function $f : V \to \{1, 2, \ldots, k\}$ such that if $e = \{v, w\} \in E$, then $f(v) \neq f(w)$.

Proposition 4.3.2. A plane graph cannot be properly colored if any only if it has a cut edge.

Definition 4.3.3. Let $G = (V, E)$ be a graph properly embedded in the plane. For each face F of G, pick a point F^* in F. For each edge e of G, let F_1 and F_2 be the faces bordering γ_e. Draw a curve e^* from F_1^* to F_2^* that intersects the graph G at a single point that is in γ_e. This can be done so that the points F^* and curves e^* also form a graph properly embedded in the plane. The unique such graph is termed the planar dual of G, and is denoted G^*.

Definition 4.3.4. The minimum value of k (with respect to the coloring function) for which G has a proper coloring is the chromatic number of G, denoted by $\chi(G)$.

Definition 4.3.5. Let $G = (V, E)$ be a planar graph. List the vertices $L : v_1, v_2, \ldots, v_p$ in some order. Let $d(v_i)$ be the number of neighbors v_i has among its predecessors. Let $t(L)$ be the maximum of $d(v_i)$ over all i. Define $\theta(G)$ to be the minimum of $t(L)$ over all ways L of listing the vertices. This number is then termed the online width of G.

Proposition 4.3.6. $\chi(G) \leqslant \theta(G) + 1$

Definition 4.3.7. For $G = (V, E)$ planar, define $\Delta(G)$ to be the maximum degree of any vertex of G. Trivially we have that $\theta(G) \leqslant \Delta(G)$.

Theorem 4.3.8. Let $G = (V, E)$ be a planar graph. Then G has a vertex of degree at most 5.

Theorem 4.3.9. Every planar graph G has $\theta(G) \leqslant 5$.

Theorem 4.3.10. [Five-color theorem]
Every planar graph may be properly colored with at most 5 colors. Or, for any G planar, $\chi(G) \leqslant 5$.

5 Bipartite graphs

5.1 Matchings

Definition 5.1.1. A matching in a graph $G = (V, E)$ is a set M of edges $M \subseteq E$ such that (V, M) is a spanning subgraph of G of maximum degree $\leqslant 1$.

To clarify: In a matching, every vertex $v \in V$ is incident with at most one edge in M.

Definition 5.1.2. A perfect matching is a spanning matching, or a 1-regular spanning subgraph.

Definition 5.1.3. A vertex-cover in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that each edge of G is incident with at least one vertex in S.

Lemma 5.1.4. Let $G = (V, E)$ be a graph, M a matching in G, and S a vertex-cover in G. Then $|M| \leqslant |S|$.

Remark 5.1.5. Let $G = (V, E)$ be a graph, M a matching in G, and S a vertex-over in G. If $|M| = |S|$, then M is a maximum size matching and S is a minimum size vertex-cover.

Theorem 5.1.6.$*$ [König's theorem]
Let $G = (V, E)$ be a bipartite graph, M a maximum matching in G, and S a minimum vertex-over in G. Then $|M| = |S|$.

5.2 Alternating paths

Definition 5.2.1. Let $G = (V, E)$ be a graph, and M a matching in G. A vertex $v \in V$ is M-saturated if it is incident with at least one edge of M, and M-unsaturated if it not.

Definition 5.2.2. A path $P : v_0 e_1 v_1 \ldots e_k v_k$ is M-alternating if every second edge is in M, and every second edge is not in M.

Definition 5.2.3. An M-augmented path is an M-alternating path with first and last vertex both M-unsaturated and not the same vertex.

Definition 5.2.4. The symmetric difference of two sets of vertices A, B is

$$A \triangle B = (A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B)$$

Lemma 5.2.5. Let $G = (V, E)$ be a graph, M a matching, and P and augmented path.
Then $M' = M \triangle E(P)$ is a matching in G with $|M'| = 1 + |M|$, so M is not a maximum matching.

Observation 5.2.6. Construct a bipartite graph $G = (V, E)$ with bipartition (A, B). Then
Let $X_o \subseteq A$ be the set of M-unsaturated vertices in A.
Let $X \subseteq A$ be the set of vertices in A connected to some $x_o \in X_o$ by an M-alternating path.
Let $Y \subseteq B$ be the set of vertices in B connected to some $x_o \in X_o$ by an M-alternating path.
Then the following three observations may be made:

1. There are no edges of G between X and $B \setminus Y$.
2. There are no edges of M between Y and $A \setminus X$.
3. If M is a maximum matching, then every vertex in Y is M-saturated.

Definition 5.2.7. Let $G = (V, E)$ be a bipartite graph with bipartition $A \cup B = V$. A matching M such that it contains every vertex in A is an A-saturating matching.

Definition 5.2.8. For a graph $G = (V, E)$ with bipartition $A \cup B = V$, for $S \subseteq A$, define the neighbors of S by the set $N(S) = \{w \in B \,\big|\, \{v, w\} \in E \text{ for some } v \in S\}$.

Remark 5.2.9. [Hall's condition]
If a graph $G = (V, E)$ has an A-saturating matching given a bipartition $A \cap B = V$, then for all $S \subseteq A$, $|N(S)| \geqslant |S|$.

Theorem 5.2.10. [Hall's theorem]
A graph $G = (V, E)$ with bipartition $A \cap B = V$ has an A-saturating matching if and only if it satisfies Hall's condition.

Corollary 5.2.11. If $k \geqslant 1$, then any k-regular bipartite graph has a perfect matching.

File III

Identities

$$\frac{1}{(1-x)^m} = \sum_{k=0}^{\infty} \binom{m+k-1}{m-1} x^k \qquad \binom{a+b}{a} = \binom{a+b}{b}$$

$$\lim_{n\to\infty} \left[\left(1+\frac{x}{n}\right)^n\right] = e^x$$

File IV
Selected proofs

Theorem 5.1.5. [König's theorem]
Let $G = (V, E)$ be a bipartite graph, M a maximum matching in G, and S a minimum vertex-over in G. Then $|M| = |S|$.

Proof: Let G have a bipartition (A, B) and M be a maximum matching in G.

Let $X_o \subseteq A$ be the set of M-unsaturated vertices in A.

Let $X \subseteq A$ be the set of vertices in A connected to some $x_o \in X_o$ by an M-alternating path.

Let $Y \subseteq B$ be the set of vertices in B connected to some $x_o \in X_o$ by an M-alternating path.

Note that $X = Y \cup (A \setminus X)$ is a vertex cover of G.

Every vertex in S is M-unsaturated, since $X_o \subseteq A$ and by observation 3.

By observation 2, there is no edge of M with both ends in S.

Every edge of M has exactly one end in S and every vertex in S has exactly one edge in M.

So there is a one-to-one and onto correspondence between M and S: every $e \in M$ is matched with a $v \in S$ if and only if $v \in e$.

Hence $|S| = |M|$.

File V

Algorithms

All algorithms are in Python.
All input graphs are of form $G = [[x_1, y_1], [x_2, y_2], ..., [x_n, y_n]]$ for vertices x_i and y_i.

Algorithm 5.1.1. [Search tree algorithm]
Input: A set of edges E of a graph $G = (V, E)$, a vertex $v \in V$.
Output: A spanning tree for the component containing v.

```
def st(G,v):
    W = [v]
    F = []
    D = []
    while 1==1:
        for i in G:
            if (i[0] in W and i[1] not in W) or (i[1] in W and i[0] not in W):
                D.append(i)
        if D == []:
            return F
        else:
            if D[0][0] not in W:
                W.append(D[0][0])
                F.append(D[0])
            else:
                W.append(D[0][1])
                F.append(D[0])
            D = []
```

Algorithm 5.1.2. [Distance computing algorithm]
This algorithm is a special case of the search tree algorithm. It uses Prim's algorithm and the BFS.
Input: A set of edges E of a graph $G = (V, E)$, a root vertex $v \in V$.
Output: A spanning tree rooted at v for the component containing v.

```
def d(G,v):
   W = [v]
   F = []
   Q = [v]
   D = []
   while len(Q) != 0:
       s = Q[0]
       for i in G:
           if s == i[0] and i[1] not in W:
               D.append(i)
           elif s == i[1] and i[0] not in W:
               D.append(i)
       while len(D) != 0:
           for j in D:
               if s == j[0]:
                   W.append(j[1])
                   Q.append(j[1])
               elif s == j[1]:
                   W.append(j[0])
```

```
                Q.append(j[0])
            F.append(j)
            D.remove(j)
    Q.remove(s)
return F
```