Math 530 Spring 2025, Lecture diary

website: `https://www.cip.ifi.lmu.de/~grinberg/t/25s`

Note: This is a rough, unedited version of what I typed in class (but lacking the illustrations I drew on the blackboard)! See the notes for a more detailed and fleshed-out writeup of this material.

0.1. Plan

This is a course on **graphs** – a rather elementary concept (actually a cluster of related concepts) that appears all over mathematics. We will discuss several kinds of graphs: simple graphs, multigraphs, simple digraphs, multidigraphs ("di" means "directed") and study their features and properties. In particular, we will see walks, cycles, paths, matchings, flows, ... on graphs.

The theory of graphs goes back to Euler in 1736. In the 19th century (?), Jacobi, Cayley, Kirchhoff, Borchardt and others picked up the subject in earnest. In the 20th century, it became mainstream. Many textbooks, lecture notes, journals on it now exist.

We will mostly follow my lecture notes from 2022 (`https://www.cip.ifi.lmu.de/~grinberg/t/22s/`). Feel free to interject with questions and ideas. If you are interested in research, this is a great place to start!

A few **administrativa**:

- The website (`https://www.cip.ifi.lmu.de/~grinberg/t/25sg`) is the syllabus.
- We will use gradescope for HW. Please sign up there, using the code I sent out.
- **Homeworks should be typewritten, not handwritten.** You can use LaTeX or Office or Google Docs or even .txt. You can scan pictures (e.g., graphs).
- Office hours: Mon 3–4 PM.

0.2. Notations

- We let $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. In particular, $0 \in \mathbb{N}$.
- The size (= cardinality) of a set S is denoted $|S|$.
- If S is a set, then the **powerset** of S means the set of all subsets of S. This powerset is denoted by $\mathcal{P}(S)$.
- Moreover, if S is a set and $k \in \mathbb{N}$, then $\mathcal{P}_k(S)$ means the set of all k-element subsets of S. For example,

$$\mathcal{P}_2(\{1,2,3\}) = \{\{1,2\}, \{1,3\}, \{2,3\}\}.$$

- For any number n and any $k \in \mathbb{N}$, we define the **binomial coefficient** $\binom{n}{k}$ to be the number

$$\frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}.$$

This is the number of k-element subsets of any given n-element set. In other words, if S is an n-element set, then $|\mathcal{P}_k(S)| = \dbinom{n}{k}$.

If $n, k \in \mathbb{N}$ and $n \geq k$, then $\dbinom{n}{k} = \dfrac{n!}{k! \cdot (n-k)!}$.

In particular, if S is an n-element set, then

$$|\mathcal{P}_2(S)| = \binom{n}{2} = \frac{n(n-1)}{2} = 1 + 2 + \cdots + (n-1).$$

Famously, the binomial coefficients satisfy **Pascal's recursion**

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

1. Simple graphs

1.1. Definition

The first type of graphs we will consider are the so-called "simple graphs":

Definition 1.1.1. A **simple graph** is a pair (V, E), where V is a finite set and E is a subset of $\mathcal{P}_2(V)$.

Thus, a simple graph is a pair (V, E), where V is a finite set, and E is a set consisting of 2-element subsets of V. We will abbreviate "simple graph" as "graph" until we introduce other kinds of graphs.

Example 1.1.2. Here is a simple graph:

$$(\{1, 2, 3, 4\}, \{\{1, 3\}, \{1, 4\}, \{3, 4\}\}).$$

Example 1.1.3. For any $n \in \mathbb{N}$, we define the ***n*-th coprimality graph** to be the simple graph $\operatorname{Cop}_n := (V, E)$, where $V = \{1, 2, \ldots, n\}$ and

$$E = \{\{u, v\} \in \mathcal{P}_2(V) \mid \gcd(u, v) = 1\}.$$

The purpose of simple graphs is to encode relations on a finite set – specifically, the relations that are binary, symmetric and irreflexive. For example, Cop_n encodes the coprimality relation on the set $\{1, 2, \ldots, n\}$, except that it "forgets" the coprimality of 1 and 1 because $\{1, 1\}$ is not a 2-element subset. For another example, if V is a set of people, and

$$E = \{\{u, v\} \in \mathcal{P}_2(V) \mid u \text{ has been married to } v \text{ at some moment}\},$$

then (V, E) is a simple graph.

The following notations let us easily reference the elements of V and E when given a graph (V, E):

Definition 1.1.4. Let $G = (V, E)$ be a simple graph.

(a) The set V is called the **vertex set** of G, and is denoted by $\operatorname{V}(G)$. The elements of V are called the **vertices** (or the **nodes**) of G.

(b) The set E is called the **edge set** of G, and is denoted by $\operatorname{E}(G)$. Its elements are called the **edges** of G. We often use the notation uv for a 2-element set $\{u, v\}$. Of course, $uv = vu$.

Note that each simple graph G satisfies $G = (\operatorname{V}(G), \operatorname{E}(G))$.

(c) Two vertices u and v of G are said to be **adjacent** (to each other) if $uv \in E$. In this case, the edge uv is said to **join** u with v (or **connect** u and v); the vertices u and v are called the **endpoints** of this edge.

(d) The **neighbors** of a vertex $v \in V$ are the vertices $u \in V$ such that $vu \in E$. In other words, they are the vertices of G that are adjacent to v.

Example 1.1.5. Let G be the simple graph

$$(\{1, 2, 3, 4\}, \{\{1, 3\}, \{1, 4\}, \{3, 4\}\}).$$

Then, its vertex set and its edge set are

$$\begin{aligned} \operatorname{V}(G) &= \{1, 2, 3, 4\}, \\ \operatorname{E}(G) &= \{\{1, 3\}, \{1, 4\}, \{3, 4\}\} = \{13, 14, 34\} \end{aligned}$$

by our abuse of notation. The vertices 1 and 3 are adjacent, but the vertices 2 and 4 are not. The neighbors of 1 are 3 and 4.

1.2. Drawing graphs

Definition 1.2.1. A simple graph G can be visually represented by **drawing** it on the plane. To do so, we represent each vertex of G by a point (at which we put the name of the vertex), and then, for each edge uv of G, we draw a curve that connects the point representing u with the point representing v.

The positions of the points and the shapes of the curves can be chosen freely, as long as they don't make the graph ambiguous.

See the notes for examples. Note that sometimes the curves will cross. Keep the crossings unambiguous!

1.3. A first fact: The Ramsey number $R(3,3) = 6$

Enough definitions; let's state a first result:

Proposition 1.3.1. Let G be a simple graph with at least 6 vertices (that is, $|\mathrm{V}(G)| \geq 6$). Then, at least one of the following two statements holds:

- **Statement 1:** There exist three distinct vertices a, b, c of G such that ab, bc and ca are edges of G.
- **Statement 2:** There exist three distinct vertices a, b, c of G such that none of ab, bc and ca is an edge of G.

In other words, in a graph with ≥ 6 vertices, you can always find three (distinct) ones that are mutually adjacent or mutually non-adjacent. In other words, in any group of ≥ 6 people, you can find 3 mutual acquaintances or 3 mutual strangers (provided that the "acquaintance" relation is symmetric).

Some terminology:

Definition 1.3.2. Let G be a simple graph.

(a) A set $\{a, b, c\}$ of three distinct vertices of G is said to be a **triangle** if ab, bc and ca are edges of G.

(b) A set $\{a, b, c\}$ of three distinct vertices of G is said to be an **anti-triangle** if none of ab, bc and ca is an edge of G.

So the proposition is saying that every graph with ≥ 6 vertices contains a triangle or an anti-triangle (or both).

Example 1.3.3. **(a)** Let G be the graph (V, E), where

$$V = \{1, 2, 3, 4, 5, 6\} \qquad \text{and}$$
$$E = \{12,\ 23,\ 34,\ 45,\ 56,\ 61\}.$$

This graph G has no triangles, but it has two anti-triangles ($\{1, 3, 5\}$ and $\{2, 4, 6\}$), so the proposition holds for it.

(b) Let G be the graph (V, E), where

$$V = \{1, 2, 3, 4, 5\} \qquad \text{and}$$
$$E = \{12,\ 23,\ 34,\ 45,\ 51\}.$$

This graph G has no triangles and no anti-triangles. No surprise, since the proposition doesn't apply to it: G has only 5 vertices.

Proof of the proposition. We need to show that G has a triangle or an anti-triangle.

Choose any vertex u of G. Then, G has at least 5 vertices distinct from u. We are in one of the following two cases:

Case 1: The vertex u has at least 3 neighbors.

Case 2: The vertex u has at most 2 neighbors.

Consider Case 1. In this case, u has three neighbors p, q, r (and maybe more). If any of pq, qr, rp is an edge of G, then we have a triangle (respectively $\{u, p, q\}$ or $\{u, q, r\}$ or $\{u, r, p\}$). Otherwise, we have an anti-triangle ($\{p, q, r\}$). So we are done in Case 1.

Now consider Case 2. In this case, u has at most 2 neighbors, and thus has at least 3 non-neighbors (since G has at least 5 vertices distinct from u). So u has three non-neighbors p, q, r (and maybe more). If any of pq, qr, rp is a non-edge of G, then we have an anti-triangle ($\{u, p, q\}$ or $\{u, q, r\}$ or $\{u, r, p\}$). Otherwise, we have a triangle ($\{p, q, r\}$). So we are done in Case 2.

Thus the proof is complete. □

Remark 1.3.4. This is one of the rare propositions in mathematics that can actually be proved by brute force: It suffices to consider graphs with exactly 6 vertices, but there are only finitely many such graphs up to isomorphism (at most 2^{15}). A computer could check all of these. Nevertheless, elegant proofs that are checkable by hand are better. In fact, the idea of the above proof ends up useful in proving the following generalization:

Proposition 1.3.5. Let r and s be two positive integers. Let G be a simple graph with at least $\dbinom{r+s-2}{r-1}$ vertices. Then, at least one of the following two statements holds:

- **Statement 1:** There exist r distinct vertices of G that are mutually adjacent.
- **Statement 2:** There exist s distinct vertices of G that are mutually non-adjacent.

For $r = s = 3$, this recovers the proposition we have proved.

This more general proposition is one of the simplest **Ramsey theorems**, part of what is known as **Ramsey theory**. You can wonder whether the number $\dbinom{r+s-2}{r-1}$ is optimal, or it can be replaced by a smaller number. For $r = s = 3$, it is, but for higher r and s, it might not be. More on that next time.

1.4. Degrees

The **degree** of a vertex in a simple graph just counts how many edges contain the vertex:

Definition 1.4.1. Let $G = (V, E)$ be a simple graph. Let $v \in V$ be a vertex. Then, the **degree** of v (with respect to G) is defined to be

$$\begin{aligned}\deg v &:= (\text{the number of edges } e \in E \text{ that contain } v) \\ &= (\text{the number of neighbors of } v) \\ &= |\{u \in V \mid uv \in E\}| \\ &= |\{e \in E \mid v \in e\}| .\end{aligned}$$

Here are some basic properties of degrees in simple graphs:

Proposition 1.4.2. Let G be a simple graph with n vertices. Let v be a vertex of G. Then,

$$\deg v \in \{0, 1, \ldots, n-1\} .$$

Proof. All neighbors of v belong to the set $\mathrm{V}(G) \setminus \{v\}$, which is an $(n-1)$-element set. □

Proposition 1.4.3 (Euler 1736)**.** Let G be a simple graph. Then, the sum of the degrees of all vertices of G is twice the number of edges of G. In other words,

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

Proof. Write G as $G = (V, E)$, so that $\mathrm{V}(G) = V$ and $\mathrm{E}(G) = E$.

Now, let N be the number of all pairs $(v, e) \in V \times E$ such that $v \in e$. Now we compute N in two different ways ("double-counting"):

1. We can obtain N by comptuing, for each $v \in V$, the number of all $e \in E$ that satisfy $v \in e$, and then summing these numbers over all v. Since these numbers are just the degrees $\deg v$, the result will be $\sum_{v \in V} \deg v$.

2. We can also obtain N by comptuing, for each $e \in E$, the number of all $v \in V$ that satisfy $v \in e$, and then summing these numbers over all e. Since these numbers are all 2 (because each edge contains exactly 2 vertices), the result will be $\sum_{e \in E} 2 = 2 \cdot |E|$.

Comparing these results, we obtain $\sum_{v \in V} \deg v = 2 \cdot |E|$, qed. □

Corollary 1.4.4 (handshake lemma)**.** Let G be a simple graph. Then, the number of vertices v of G whose degree $\deg v$ is odd is even.

Proof. A sum of integers is even if and only if it has an odd number of odd addends. So $\sum_{v \in V} \deg v = 2 \cdot |E|$ yields that the sum $\sum_{v \in V} \deg v$ has an odd number of odd addends. Qed. □

Proposition 1.4.5. Let G be a simple graph with at least two vertices. Then, there exist two distinct vertices v and w of G that have the same degree.

Proof. Assume the contrary. Thus, all n vertices of G have distinct degrees (where $n = |\operatorname{V}(G)|$). In other words, the map

$$\deg : \operatorname{V}(G) \to \{0, 1, \ldots, n-1\},$$
$$v \mapsto \deg v$$

is injective. Hence, by the pigeonhole principle, this map must be bijective (since it is an injective map between two finite sets of the same size). In particular, it is surjective, so there exist a vertex u of degree 0 and a vertex v of degree $n-1$. The vertex u (having degree 0) must be adjacent to no one, whereas the vertex v (having degree $n-1$) must be adjacent to everyone. So are u and v adjacent to each other? Contradiction. (Note: u and v are distinct, since $n \geq 2$ entails $0 \neq n-1$.) □

Here is an application of neighbors to proving properties of graphs. This is known as **Mantel's theorem**:

Theorem 1.4.6 (Mantel's theorem)**.** Let G be a simple graph with n vertices and e edges, where $e > n^2/4$. Then, G has a triangle (i.e., three distinct vertices that are mutually adjacent).

Proof of Mantel's theorem. We will prove the theorem by strong induction on n. Thus, we assume (as the induction hypothesis) that the theorem holds for all graphs with fewer than n vertices. We must now prove it for our graph G with n vertices. Write G as $G = (V, E)$.

We must prove that G has a triangle. Assume the contrary.

From $e > n^2/4 \geq 0$, we see that G has an edge. Pick such an edge vw.

Now, we color each edge of G with one of three colors:

- The edge vw is colored black.
- Each edge that contains exactly one of v and w is colored green.
- All other edges are colored blue.

Let us now count (= upper-bound) the edges of each color:

- There is exactly 1 black edge.
- The number of green edges is at most $n-2$, since each vertex in $V \setminus \{v, w\}$ is adjacent to at most one of v and w (since otherwise, we would get a triangle).
- The blue edges form a graph on $n-2$ vertices with no triangles. So, by the IH, this graph has $\leq (n-2)^2/4$ edges (since having more than $(n-2)^2/4$ edges would cause a triangle by the IH).

Altogether, G has at most

$$1 + (n-2) + (n-2)^2/4 = n^2/4$$

edges, which contradicts $e > n^2/4$. Induction complete. □

Mantel's theorem can be generalized:

Theorem 1.4.7 (Turan's theorem)**.** Let r be a positive integer. Let G be a simple graph with n vertices and e edges, where $e > \frac{r-1}{r} \cdot \frac{n^2}{2}$. Then, G has $r+1$ mutually adjacent vertices (i.e., every two distinct ones among them are adjacent).

Mantel's theorem is the particular case for $r = 2$. We will prove Turan's theorem later.

1.5. Graph isomorphism

Two graphs can be distinct and yet "the same up to the names of their vertices": for instance,

$$1 - 2 - 3 \qquad \text{and} \qquad 1 - 3 - 2.$$

Let us make this formal:

Definition 1.5.1. Let G and H be two simple graphs.

(a) A **graph isomorphism** (or **isomorphism**) from G to H means a bijection $\phi : \operatorname{V}(G) \to \operatorname{V}(H)$ that "preserves edges", i.e., has the following property: For any two vertices u and v of G, we have

$$(uv \in \operatorname{E}(G)) \iff (\phi(u)\,\phi(v) \in \operatorname{E}(H)).$$

(b) We say that G and H are **isomorphic** (this is written $G \cong H$) if there exists a graph isomorphism from G to H.

In our above example, there is an isomorphism from the graph $1-2-3$ to the graph $1-3-2$ that sends $1, 2, 3$ to $1, 3, 2$, respectively. Another isomorphism between the same two graphs sends $1, 2, 3$ to $2, 3, 1$. So the graphs $1-2-3$ and $1-3-2$ are isomorphic (even in two ways).

Here are some basic properties of isomorphisms:

- If ϕ is an isomorphism from G to H, then ϕ^{-1} is an isomorphism from H to G.
- If ϕ is an isomorphism from G to H and ψ is an isomorphism from H to I, then $\psi \circ \phi$ is an isomorphism from G to I.

Graph isomorphisms preserve all "intrinsic" properties of graphs: e.g., if ϕ is an isomorphism from G to H, then

- $|\operatorname{V}(G)| = |\operatorname{V}(H)|$;
- $|\operatorname{E}(G)| = |\operatorname{E}(H)|$;
- every $v \in \operatorname{V}(G)$ satisfies $\deg_G v = \deg_H(\phi(v))$.

One use of graph isomorphisms is to relabel the vertices of a graph. For example, the vertices of any n-vertex graph can be relabelled as $1, 2, \ldots, n$ or as any other n distinct objects:

Proposition 1.5.2. Let G be a simple graph. Let S be a finite set such that $|S| = |\operatorname{V}(G)|$. Then, there exists a simple graph H that is isomorphic to G and has vertex set $\operatorname{V}(H) = S$.

Proof. Straightforward. □

1.6. Some families of graphs

We will now define some particularly significant families of graphs.

1.6.1. Complete and empty graphs

Definition 1.6.1. Let V be a finite set.

(a) The **complete graph** on V means the simple graph $(V, \mathcal{P}_2(V))$. It is the simple graph with vertex set V in which every two distinct vertices are adjacent.

If $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$, then this graph is denoted K_n.

(b) The **empty graph** on V means the simple graph $(V, \varnothing)$. It has no edges.

Note that a simple graph G is isomorphic to the complete graph K_n if and only if it has n vertices and is a complete graph (i.e., every two distinct vertices are adjacent).

Question: Given two finite sets V and W, what are the isomorphisms from the complete graph on V to the complete graph on W ?

Answer: All the bijections from V to W. If $|V| = |W|$, then there are $|V|!$ of them; otherwise there are none.

1.6.2. Path and cycle graphs

Definition 1.6.2. For each $n \in \mathbb{N}$, we define the n**-th path graph** P_n to be the simple graph

$$\left(\{1, 2, \ldots, n\}, \underbrace{\{12, 23, 34, \ldots, (n-1)\, n\}}_{= \{\{i, i+1\} \mid 1 \leq i < n\}} \right).$$

This graph has n vertices and $n-1$ edges (unless $n = 0$, in which case it has no edges).

Definition 1.6.3. For each $n > 2$, we define the n**-th cycle graph** C_n to be the simple graph

$$\left(\{1, 2, \ldots, n\}, \underbrace{\{12, 23, 34, \ldots, (n-1)\, n, n1\}}_{= \{\{i, i+1\} \mid 1 \leq i < n\} \cup \{\{n, 1\}\}} \right).$$

This graph has n vertices and n edges. We will later define C_1 and C_2 as well (these are multigraphs, not simple graphs).

Note that $C_3 = K_3$.

Question: What are the graph isomorphisms from P_n to itself?

Answer: One such isomorphism is $\mathrm{id} : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$. Another is the "flip"="reversal" map

$$\{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\},$$
$$i \mapsto n + 1 - i.$$

There are no others.

Question: What are the graph isomorphisms from C_n to itself (for $n > 2$) ?

Answer: All the rotations and the reflections. They form the dihedral group D_n (or D_{2n} depending on whom you ask). There are $2n$ of them.

1.6.3. Kneser graphs

Here is a more obscure family of graphs:

Example 1.6.4. If S is a finite set, and if $k \in \mathbb{N}$, then we define the **k-th Kneser graph of** S to be the simple graph

$$K_{S,k} := (\mathcal{P}_k(S), \{IJ \mid I, J \in \mathcal{P}_k(S) \text{ and } I \cap J = \varnothing\}).$$

So the vertices of $K_{S,k}$ are the k-element subsets of S, and two such subsets are adjacent if they are disjoint.

The Kneser graph $K_{\{1,2,3,4,5\},\ 2}$ is called the **Petersen graph**.

1.7. Subgraphs

Definition 1.7.1. Let $G = (V, E)$ be a simple graph.

(a) A **subgraph** of G means a simple graph of the form $H = (W, F)$, where $W \subseteq V$ and $F \subseteq E$. In other words, a subgraph of G means a simple graph whose vertices are vertices of G and whose edges are edges of G.

(b) Let S be a subset of V. The **induced subgraph of** G **on the set** S denotes the subgraph

$$G[S] := (S,\ E \cap \mathcal{P}_2(S))$$

of G. In other words, it denoted the subgraph of G whose vertices are the elements of S, and whose edges are precisely those edges of G whose both endpoints belong to S.

(c) An **induced subgraph** of G means a subgraph of G that is the induced subgraph of G on S for some $S \subseteq V$.

Thus, a subgraph of a graph G is obtained by throwing away some vertices and some edges of G (in such a way that no edges remain "dangling"). Such a subgraph is an induced subgraph if and only if no edges are removed without need – i.e., no edges are removed unless they lose a vertex.

So induced subgraphs can be characterized as follows:

Proposition 1.7.2. Let H be a subgraph of a simple graph G. Then, H is an induced subgraph of G if and only if each edge uv of G whose endpoints u and v are vertices of H is an edge of H.

Example 1.7.3. Let $n > 2$ be an integer. Then:

1. The path graph P_n is a subgraph of C_n. It is not induced, since it contains the vertices n and 1 but not the edge $n1$.

2. The path graph P_{n-1} is an induced subgraph of P_n.

3. Is C_{n-1} a subgraph of C_n (for $n > 3$) ? No, since C_{n-1} has the edge $(n-1)\,1$, which C_n does not have.

Note that if a subgraph of a simple graph is complete, then it is automatically an induced subgraph. In particular, a triangle in a graph is the same thing as a subgraph (or equivalently an induced subgraph) isomorphic to K_3 (or equivalently C_3). So one often says "a K_3 in G" instead of saying "a triangle in G".

1.8. Disjoint unions

A way of constructing new graphs from old is the disjoint union. The idea is simple: Taking the disjoint union $G_1 \sqcup G_2 \sqcup \cdots \sqcup G_k$ of several graphs $G_1, G_2, \ldots, G_k$ means putting the graphs alongside each other and treating the result as one big graph. To make the definition watertight, we need to rename the vertices so that they don't collide; we do this by renaming each vertex v of G_i as (i, v). Thus we define the disjoint union as follows:

Definition 1.8.1. Let $G_1, G_2, \ldots, G_k$ be simple graphs, where $G_i = (V_i, E_i)$ for each i. The **disjoint union** $G_1 \sqcup G_2 \sqcup \cdots \sqcup G_k$ of these k graphs $G_1, G_2, \ldots, G_k$ is defined to be the simple graph (V, E), where

$$V = \{(i, v) \mid i \in \{1, 2, \ldots, k\} \text{ and } v \in V_i\} \qquad \text{and}$$
$$E = \{\{(i, v_1), (i, v_2)\} \mid i \in \{1, 2, \ldots, k\} \text{ and } v_1 v_2 \in E_i\}.$$

Note: In general, $G \sqcup H$ and $H \sqcup G$ are not the same graph, just isomorphic.

1.9. Walks and paths

We now come to the definitions of walks and paths – two of the most fundamental features of graphs. In particular, Euler's 1736 paper – the first source on graphs – is devoted to certain kinds of walks.

1.9.1. Definitions

Imagine a graph as a road network, where each vertex is a town and each edge is a (bidirectional) road. By successively walking along several edges, you can get from one town to another even if they are not adjacent. This is made formal in the notion of a "walk". There is a related notion of a "path", which is a walk that does not revisit any vertex; a notion of a "circuit", which is a closed walk (i.e., its destination is its origin); and a notion of a "cycle", which is a circuit that revisits no vertices other than the origin at the end. Now we will give definitions and study these notions.

Definition 1.9.1. Let G be a simple graph. Then:

(a) A **walk** (in G) means a finite sequence $(v_0, v_1, \ldots, v_k)$ of vertices of G (with $k \geq 0$) such that all of $v_0v_1,\ v_1v_2,\ v_2v_3,\ \ldots,\ v_{k-1}v_k$ are edges of G. (The latter condition is vacuously true if $k = 0$.)

(b) If $\mathbf{w} = (v_0, v_1, \ldots, v_k)$ is a walk in G, then:

- The **vertices** of $\mathbf{w}$ are $v_0, v_1, \ldots, v_k$.
- The **edges** of $\mathbf{w}$ are $v_0v_1,\ v_1v_2,\ \ldots,\ v_{k-1}v_k$.
- The **length** of $\mathbf{w}$ is k (that is, the # of edges, or the # of vertices minus 1; everything here is counted with multiplicity).
- The **starting point** of $\mathbf{w}$ is v_0, and we say that $\mathbf{w}$ **starts** at v_0.
- The **ending point** of $\mathbf{w}$ is v_k, and we say that $\mathbf{w}$ **ends** at v_k.

(c) A **path** (in G) means a walk (in G) whose vertices are distinct. In other words, a walk $(v_0, v_1, \ldots, v_k)$ is a path if and only if $v_0, v_1, \ldots, v_k$ are distinct.

(d) Let p and q be two vertices of G. A **walk/path from** p **to** q means a walk/path that starts at p and ends at q.

(e) We often say "walk of G" and "path of G" instead of "walk in G" and "path in G".

Note that the edges of a path are always distinct.

1.9.2. Composing/concatenating and reversing walks

We can "splice" two walks together if the ending point of the first is the starting point of the second:

Proposition 1.9.2. Let G be a simple graph. Let u, v, w be three vertices. Let $\mathbf{a} = (a_0, a_1, \ldots, a_k)$ be a walk from u to v. Let $\mathbf{b} = (b_0, b_1, \ldots, b_\ell)$ be a walk from v to w. Then,

$$\begin{aligned} \mathbf{a} * \mathbf{b} &:= (a_0, a_1, \ldots, a_{k-1}, b_0, b_1, \ldots, b_\ell) \\ &= (a_0, a_1, \ldots, a_k, b_1, b_2, \ldots, b_\ell) \\ &= (a_0, a_1, \ldots, a_{k-1}, v, b_1, b_2, \ldots, b_\ell) \end{aligned}$$

is a walk from u to w.

Proof. Intuitively clear and very easy. □

Note, however, that if $\mathbf{a}$ and $\mathbf{b}$ are paths, then $\mathbf{a} * \mathbf{b}$ might not be a path.

Proposition 1.9.3. Let G be a simple graph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, a_1, \ldots, a_k)$ be a walk from u to v. Then,

$$\operatorname{rev} \mathbf{a} := (a_k, a_{k-1}, \ldots, a_0)$$

is a walk from v to u. Moreover, if $\mathbf{a}$ is a path, then so is $\operatorname{rev} \mathbf{a}$.

Proof. Intuitively clear and very easy. □

1.9.3. Reducing walks to paths

A path is just a walk without repeated vertices. If you have a walk, then you can turn it into a path by removing such "loops":

Proposition 1.9.4. Let G be a simple graph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, a_1, \ldots, a_k)$ be a walk from u to v. Assume that $\mathbf{a}$ is not a path. Then, there exists a walk from u to v whose length is smaller than k.

Proof. Since $\mathbf{a}$ is not a path, two of its vertices are equal. In other words, there exist $i < j$ such that $a_i = a_j$. Now, the tuple

$$(a_0, a_1, \ldots, a_i, a_{j+1}, a_{j+2}, \ldots, a_k)$$

is a walk from u to v whose length is smaller than k. □

Corollary 1.9.5 (When there is a walk, there is a path)**.** Let G be a simple graph. Let u and v be two vertices of G. Assume that there is a walk from u to v of length k for some $k \in \mathbb{N}$. Then, there is a path from u to v of length $\leq k$.

Proof. Apply the preceding proposition repeatedly, until you have a path. □

Note that in a simple graph with n vertices, the highest possible length of a path is $n-1$ (since all its vertices must be distinct, and so there cannot be more than n of them). So there are only finitely many possible paths in a graph. In contrast, walks can be arbitrarily long, and there are usually infinitely many of them.

1.9.4. The equivalence relation "path-connected"

Definition 1.9.6. Let G be a simple graph. We define a binary relation $\simeq_G$ on the set $\operatorname{V}(G)$ as follows: For two vertices u and v, we write $u \simeq_G v$ if and only if there exists a walk from u to v in G. By the corollary above, this is equivalent to the existence of a path from u to v in G.

This binary relation $\simeq_G$ is called "**path-connectedness**" or just "**connectedness**". When two vertices u and v satisfy $u \simeq_G v$, we say that u and v are **path-connected**.

Proposition 1.9.7. Let G be a simple graph. Then, the relation $\simeq_G$ is an equivalence relation.

Proof. Follows from the properties shown above. □

Definition 1.9.8. Let G be a simple graph. The equivalence classes of the path-connectedness relation $\simeq_G$ are called the **connected components** (or, short, **components**) of G.

We say that G is **connected** if G has exactly one component.

Note: The empty graph with no vertices at all has 0 components, not exactly one component, so it is not connected!

A complete graph K_n with n vertices is connected if and only if $n \neq 0$.

It would be bad if the following was not true:

Proposition 1.9.9. Let G be a simple graph. Let C be a component of G. Then, the induced subgraph of G on the subset C is connected. (In short: A connected component is connected.)

Proof. In a nutshell, we must argue that any two vertices u and v in C must be path-connected not just by a path of G, but actually by a path of the induced subgraph of G on C. But this is easy: Any path from u to v must stay within C, because any vertex on this path is path-connected to u and therefore lies in the same component as u, which is C. Thus, any path from u to v is a path in the induced subgraph.

(More details in the notes.) □

In the following proposition, we are using the notation $G[S]$ for the induced subgraph of a simple graph G on a subset S of its vertex set.

Proposition 1.9.10. Let G be a simple graph. Let $C_1, C_2, \ldots, C_k$ be all its components (with no repetition). Then,

$$G \cong G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k].$$

Proof. Consider the bijection from $\operatorname{V}(G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k])$ to $\operatorname{V}(G)$ that sends each vertex (i, v) to v. This bijection is a graph isomorphism, since there are no edges of G that join vertices from different components. □

The upshot of these results is that every simple graph can be decomposed into a disjoint union of connected graphs (its components, or, more precisely, its induced subgraphs on its component). This decomposition is essentially unique.

1.10. Closed walks and cycles

Here are two further kinds of walks:

Definition 1.10.1. Let G be a simple graph.

- **(a)** A **closed walk** of G means a walk whose first vertex is identical with its last vertex. In other words, it means a walk $(w_0, w_1, \ldots, w_k)$ with $w_0 = w_k$. Some call this a **circuit**.
- **(b)** A **cycle** of G means a closed walk $(w_0, w_1, \ldots, w_k)$ such that $k \geq 3$ and such that the vertices $w_0, w_1, \ldots, w_{k-1}$ are distinct.

Authors have different opinions about whether $(1, 2, 3, 1)$ and $(2, 3, 1, 2)$ and $(1, 3, 2, 1)$ are different cycles or the same cycle.

Are paths and cycles related to path graphs P_n and cycle graphs C_n ? Yes:

Proposition 1.10.2. Let G be a simple graph.

- **(a)** If $(p_0, p_1, \ldots, p_k)$ is a path of G, then there is a subgraph of G isomorphic to P_{k+1}, namely the subgraph

$$(\{p_0, p_1, \ldots, p_k\}, \{p_i p_{i+1} \mid 0 \leq i < k\}).$$

  Conversely, any subgraph of G isomorphic to P_{k+1} gives a path of G.

(b) Now assume that $k \geq 3$. If $(c_0, c_1, \ldots, c_k)$ is a cycle of G, then there is a subgraph of G isomorphic to C_k, namely

$$(\{c_0, c_1, \ldots, c_k\}, \{c_i c_{i+1} \mid 0 \leq i < k\}).$$

Conversely, any subgraph of G isomorphic to C_k gives a cycle of G.

Proof. Straightforward. □

Certain graphs contain cycles; others don't. Let us find some criteria for when a graph can and when it cannot have cycles:

Definition 1.10.3. A walk $\mathbf{w} = (w_0, w_1, \ldots, w_k)$ in a simple graph G is called **backtrack-free** if there exists no i such that $w_{i-1}w_i = w_i w_{i+1}$. (In other words, if no two consecutive edges of $\mathbf{w}$ are identical.)

Proposition 1.10.4. Let G be a simple graph. Let $\mathbf{w}$ be a backtrack-free walk of G. Then, $\mathbf{w}$ either is a path or contains a cycle.

Proof. Write $\mathbf{w}$ as $\mathbf{w} = (w_0, w_1, \ldots, w_k)$. Assume that $\mathbf{w}$ is not a path. We must show that $\mathbf{w}$ contains a cycle.

Since $\mathbf{w}$ is not a path, there exist $i < j$ such that $w_i = w_j$. Among all such pairs (i, j), pick one with **minimum** difference $j - i$.

We shall show that $(w_i, w_{i+1}, \ldots, w_j)$ is a cycle. For this purpose, we need to show that all the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct, and that $j - i \geq 3$.

To prove the distinctness, we observe that if two of $w_i, w_{i+1}, \ldots, w_{j-1}$ are equal, say $w_u = w_v$ with $i \leq u < v < j$, then (u, v) would be an equal-vertex pair with smaller difference $v - u$ than $j - i$, which would contradict the minimality of $j - i$. So $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct.

To prove that $j - i \geq 3$, we note that $j - i = 2$ is impossible (since that would mean that $w_i w_{i+1} w_j$ is a backtrack, but $\mathbf{w}$ is backtrack-free), and $j - i = 1$ is impossible as well (since $w_i w_i$ is not an edge).

So we conclude that $(w_i, w_{i+1}, \ldots, w_j)$ is a cycle, and the proof is complete. □

Corollary 1.10.5. Let G be a simple graph that has a closed backtrack-free walk of length > 0. Then, G has a cycle.

Proof. By the previous proposition, this walk must be a path or contain a cycle. But it cannot be a path, since it is closed and has length > 0. □

Theorem 1.10.6. Let G be a simple graph. Let u and v be two vertices in G. Assume that there are two distinct backtrack-free walks from u to v. Then, G has a cycle.

Proof. Let $\mathbf{p} = (p_0, p_1, \ldots, p_k)$ and $\mathbf{q} = (q_0, q_1, \ldots, q_\ell)$ be the two distinct backtrack-free walks from u to v. We can consider the walk

$$\mathbf{p} * \operatorname{rev} \mathbf{q} = (u = p_0, p_1, \ldots, p_k = v = q_\ell, q_{\ell-1}, \ldots, q_0 = u);$$

this is a closed walk of length > 0. If we can show that this walk $\mathbf{p} * \operatorname{rev} \mathbf{q}$ is backtrack-free, then we are done by the previous corollary.

Sadly, $\mathbf{p} * \operatorname{rev} \mathbf{q}$ is not always backtrack-free. But the only case when it fails to be so is when $\mathbf{p}$ and $\mathbf{q}$ have the same last edge. In this case, we can reduce $\mathbf{p}$ and $\mathbf{q}$ to two shorter walks by removing this last edge. Thus we get two shorter distinct backtrack-free walks from u to the same ending point v'. Hence, we can proceed by induction on the length of one of our walks.

(See the notes for details.) □

1.11. The longest path trick

The following nice proposition illustrates a useful tactic in dealing with graphs:

> **Proposition 1.11.1.** Let G be a simple graph with at least one vertex. Let $d > 1$ be an integer. Assume that each vertex of G has degree $\geq d$. Then, G has a cycle of length $\geq d + 1$.

Proof. Let $\mathbf{p} = (v_0, v_1, \ldots, v_m)$ be a **longest** path of G. (Why does such a thing exist? Any path of G has length $\leq |V| - 1$, since its vertices must be distinct. Also, G has at least one vertex and therefore at least one path.)

The vertex v_0 has degree $\geq d$ (by assumption), and thus has $\geq d$ many neighbors.

If all neighbors of v_0 belonged to the set $\left\{v_1, v_2, \ldots, v_{\min\{d-1,m\}}\right\}$, then v_0 would have at most $d - 1$ many neighbors, which would contradict the previous sentence. Thus, v_0 must have a neighbor u that does not belong to this set. Consider this u.

Attaching the vertex u to the front of the path $\mathbf{p}$, we obtain a walk

$$\mathbf{p}' := (u, v_0, v_1, \ldots, v_m).$$

This walk cannot be a path, since $\mathbf{p}$ is already a longest path of G. Thus, u must be one of the vertices $v_0, v_1, \ldots, v_m$. But u cannot be v_0 (since it is a neighbor of v_0) and also cannot be among the first $d - 1$ of the vertices $v_1, v_2, \ldots, v_m$ (since we just showed that u does not belong to the set $\left\{v_1, v_2, \ldots, v_{\min\{d-1,m\}}\right\}$). Hence, we must have

$$u = v_k \qquad \text{for some } k \geq d.$$

Thus, the walk $(u, v_0, v_1, \ldots, v_k)$ is a cycle of length $k + 1 \geq d + 1$, as desired. □

1.12. Bridges

What happens to a graph if we remove a single edge from it? This is a crucial question, simple as it is. Let us first define a notation for this:

Definition 1.12.1. Let $G = (V, E)$ be a simple graph. Let e be an edge of G. Then, $G \setminus e$ will mean the graph obtained from G by removing the edge e, that is,

$$G \setminus e = (V,\ E \setminus \{e\}) .$$

Some authors denote this graph by $G - e$.

Theorem 1.12.2. Let G be a simple graph. Let e be an edge of G. Then:

- **(a)** If e is an edge of some cycle of G, then the components of $G \setminus e$ are precisely the components of G. (Keep in mind that the components are sets – they are the equivalence classes of the path-connectedness relation.)
- **(b)** If e appears in no cycle of G, then the graph $G \setminus e$ has one more component than G.

Proof. This is an outline; see a reference in the notes for details.

Let u and v be the endpoints of the edge e, so that $e = uv$. Note that $u \simeq_G v$, since (u, v) is a path of G.

(a) Assume that e is an edge of some cycle of G. Then, there is a path from u to v in $G \setminus e$ by walking around the rest of this cycle. Therefore, $u \simeq_{G \setminus e} v$.

Now, we must show that the components of $G \setminus e$ are precisely the components of G. This will clearly follow if we can show that the relations $\simeq_{G \setminus e}$ and $\simeq_G$ are the same. In other words, we must show that two vertices x and y satisfy $x \simeq_{G \setminus e} y$ if and only if $x \simeq_G y$. The "only if" part is clear (every walk of $G \setminus e$ is a walk of G). For the "if" part, note that any walk in G can be transformed into a walk in $G \setminus e$ by replacing the edge e with the rest of the cycle on which it lies. So we are done with part **(a)**.

(b) Assume that e appears in no cycle of G. We must prove that $G \setminus e$ has one more component than G. To do so, it suffices to show the following:

> *Claim 1:* The component of G that contains u and v (this is a single component since $u \simeq_G v$) breaks into two components of $G \setminus e$ when the edge e is removed.
>
> *Claim 2:* All other components of G remain components of $G \setminus e$.

Claim 2 is pretty clear: The components of G that don't contain u and v do not change at all when e is removed (since they contain neither endpoint of u).

It remains to prove Claim 1. We introduce some notations:

- Let C be the component of G that contains u and v.
- Let A be the component of $G \setminus e$ that contains u.
- Let B be the component of $G \setminus e$ that contains v.

We must show that $A \cup B = C$ and $A \cap B = \varnothing$.

Let us first prove that $A \cap B = \varnothing$. In other words, we need to show that $u \simeq_{G \setminus e} v$ does **not** hold. In other words, we must show that there is no path from u to v in $G \setminus e$. But if there was such a path, then we could "close" this path to a cycle of G by appending the edge e to it. This would contradict the assumption that e is not contained in any cycle. So we have proved $A \cap B = \varnothing$.

Now let us prove that $A \cup B = C$. Easily, $A \cup B \subseteq C$ (because every walk of $G \setminus e$ is a walk of G), so we only need to prove $C \subseteq A \cup B$.

Let $c \in C$. This means that $c \simeq_G u$, so there is a path from c to u in G. Now two cases are possible:

- If this path does not use the edge e, then it is still a path in $G \setminus e$, so we get $c \simeq_{G \setminus e} u$ and therefore $c \in A$.
- If this path does use the edge e, then this path must arrive at v before this edge (otherwise, it would arrive at u and therefore contain u twice, which is not allowed for a path), and therefore an appropriate piece of this path joins c to v in $G \setminus e$, which shows that $c \in B$.

In either case, $c \in A \cup B$. Since we have shown this for all $c \in C$, we thus have shown that $C \subseteq A \cup B$. This completes our proof. □

Definition 1.12.3. Let e be an edge of a simple graph G.

(a) We say that e is a **bridge** of G if e appears in no cycle of G.

(b) We say that e is a **cut-edge** of G if the graph $G \setminus e$ has more components than G.

Corollary 1.12.4. Let e be an edge of a simple graph G. Then, e is a bridge if and only if e is a cut-edge.

We can also define **cut-vertices**: Vertices whose removal breaks the graph into more components than it had. But these aren't as well-behaved.

1.13. Dominating sets

Here is another concept we can define for a graph:

Definition 1.13.1. Let $G = (V, E)$ be a simple graph.

A subset U of V is said to be **dominating** (for G) if each vertex $v \in V \setminus U$ has at least one neighbor in U.

A **domating set** for G means a subset of V that is dominating.

Exercise 1. Consider the cycle graph C_5. Here are some dominating sets:

$$\{1,3,5\}, \qquad \{1,2,3,4,5\}, \qquad \{2,4\}.$$

Here are some non-dominating sets:

$$\{1\}, \qquad \{1,2\}.$$

In general, any subset with ≥ 3 vertices is dominating, whereas any subset with ≤ 1 vertex is not. Two-vertex subsets are dominating if and only if they are not edges.

Some more examples:

- If $G = (V, E)$ is a simple graph, then the whole vertex set V is always dominating, whereas $\varnothing$ is dominating if and only if $V = \varnothing$,
- If $G = (V, E)$ is a complete graph, then any nonempty subset of V is dominating.
- If $G = (V, E)$ is an empty graph, then only V is dominating.

Clearly, the "denser" a graph is, the "easier" it is for a subset of V to be dominating. We shall now make this more concrete by giving some criteria.

First, a definition:

Definition 1.13.2. Let G be a simple graph. A vertex v of G is said to be **isolated** if $\deg v = 0$.

An isolated vertex must belong to every dominating set, but otherwise does not affect the domination. So when studying dominating sets, we can get rid of all isolated vertices beforehand.

Proposition 1.13.3. Let $G = (V, E)$ be a simple graph that has no isolated vertices. Then:

(a) There exists a dominating set of G that has size $\leq |V|/2$.

(b) There exist two disjoint dominating sets A and B of G such that $A \cup B = V$.

One proof of this is on the homework.

Next, we state a rather surprising result that is also fairly recent (Brouwer, 2009):

Theorem 1.13.4 (Brouwer's dominating set theorem)**.** Let G be a simple graph. Then, the number of dominating sets of G is odd.

Brouwer gives three proofs. Here is my favorite one. First, a notation:

Definition 1.13.5. Let $G = (V, E)$ be a simple graph. A **detached pair** will mean a pair (A, B) of two subsets A and B of V such that there exist no edge $ab \in E$ with $a \in A$ and $b \in B$.

Note: Pair = ordered pair. So if (A, B) is a detached pair, then so is (B, A) (and this is a different detached pair unless $A = B = \varnothing$).

Let's now try to prove Brouwer's theorem:

Proof of Brouwer's theorem. Write G as $G = (V, E)$. Recall that $\mathcal{P}(V)$ denote the set of all subsets of V.

Construct a new graph H as follows:

- The vertices of H are the subsets of V. (So its vertex set is $\mathcal{P}(V)$.)
- Two subsets A and B of V are adjacent in H if and only if (A, B) is a detached pair.

I claim that the vertices of H that have odd degree are precisely the subsets of V that are dominating. In other words:

Claim 1: Let A be a subset of V. Then, A has odd degree in H if and only if A is a dominating set of G.

Proof of Claim 1. The degree of A in H is the number of subsets B of V such that (A, B) is a detached pair, i.e., such that B contains neither an element of A nor a neighbor of an element of A. In other words, it is the number of subsets of $V \setminus (A \cup N(A))$, where

$$N(A) = \{\text{neighbors of } A\} = \{v \in V \mid v \text{ has a neighbor in } A\}.$$

So it equals $2^{|V \setminus (A \cup N(A))|}$. This is odd if and only if $|V \setminus (A \cup N(A))| = 0$, which is saying precisely that $A \cup N(A) = V$, which in turn means that A is dominating for G. So Claim 1 is proved. □

Claim 1 shows that the vertices of H that have odd degree are precisely the dominating sets of G. But the handshake lemma says that the # of odd-degree vertices in a simple graph is even. So we conclude that the # of dominating sets of G is even.

Wait, we were trying to prove that it is odd!

The problem with our above argument lies in the definition of H: "Two subsets A and B of V are adjacent in H if and only if (A, B) is a detached pair" only works if A and B are distinct, since we want H to be a simple graph. Thus, we cannot count the detached pair $(\varnothing, \varnothing)$. So Claim 1 holds for any nonempty A, but not for $A = \varnothing$. Fixing the $A = \varnothing$ case by hand, we arrive at the following corrected version of Claim 1:

> *Claim 1′:* Let A be a subset of V, and assume that $V \neq \varnothing$. Then, A has odd degree in H if and only if A is empty or a dominating set of G.

Thus, the handshake lemma now yields that there is an even number of subsets of V that are empty or dominating. But the empty set is not dominating, so we can subtract 1 and get the number of dominating subsets. And so this number is odd. □

A particularly nice proof was recently (2017) found by Irene Heinrich and Peter Tittmann.; they gave an "explicit" formula for the number of dominating sets that shows that this number is odd:

Theorem 1.13.6 (Heinrich–Tittmann formula)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n > 0$.

Let α be the number of all detached pairs (A, B) such that both $|A|$ and $|B|$ are even and positive.

Let β be the number of all detached pairs (A, B) such that both $|A|$ and $|B|$ are odd.

Then, α and β are even, and the number of dominating sets of G is $2^n - 1 + \alpha - \beta$.

See the references in the notes for how to prove this.

1.14. Hamiltonian paths and cycles

1.14.1. Basics

Quick question: Given a simple graph G, when is there a closed **walk** that contains each vertex of G ? Precisely when G is connected.

The question becomes a lot more interesting if we replace "closed walk" by "path" or "cycle". These objects have a name:

Definition 1.14.1. Let $G = (V, E)$ be a simple graph.

(a) A **Hamiltonian path** (short: **hamp**) in G means a walk of G that contains each vertex of G exactly once. Obviously, it is a path.

(b) A **Hamiltonian cycle** (short: **hamc**) in G means a cycle $(v_0, v_1, \ldots, v_k)$ of G such that each vertex of G appears exactly once among $v_0, v_1, \ldots, v_{k-1}$.

Some graphs have Hamiltonian paths; some don't.

In general, finding a hamp or hamc, or proving that none exists, is a finite but hard problem. There are algorithms with running time $O\left(n^2 2^n\right)$, where n

is the number of vertices of the graph. The problem is NP-hard (in both hamp and hamc versions) and a focus of ongoing research (particular cases, improved algorithms, variants such as the travelling salesman problem).

1.14.2. Sufficient criteria: Ore and Dirac

We shall now show some necessary criteria and some sufficient criteria (but not both at once) for the existence of hamps and hamcs. The most famous sufficient criterion is the following:

Theorem 1.14.2 (Ore)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n \geq 3$.

Assume that $\deg x + \deg y \geq n$ for any two non-adjacent distinct vertices x and y.

Then, G has a hamc.

Proof. A **listing** (of V) shall mean a list of elements of V that contains each element exactly once. Clearly, it is an n-tuple.

The **hamness** of a listing $(v_1, v_2, \ldots, v_n)$ means the number of all $i \in \{1, 2, \ldots, n\}$ such that $v_i v_{i+1} \in E$, where $v_{n+1} := v_1$. Clearly, a hamc is a listing with hamness n. Note that the hamness of a listing does not change if we cyclically rotate it.

So how do we find a listing with hamness n ? We start with any listing (there are $n!$ of them), and we gradually increase its hamness. For this we need the following:

> *Claim 1:* Let $(v_1, v_2, \ldots, v_n)$ be a listing of hamness $k < n$. Then, there exists a listing of hamness larger than k.

Proof of Claim 1. Since $(v_1, v_2, \ldots, v_n)$ has hamness $< n$, there exists some $i \in \{1, 2, \ldots, n\}$ such that $v_i v_{i+1} \notin E$. Pick such an i. Then, the vertices v_i and v_{i+1} are not adjacent. Hence, the assumption of the theorem yields $\deg(v_i) + \deg(v_{i+1}) \geq n$.

This entails by the pigeonhole principle that there exists a $j \in \{1, 2, \ldots, n\} \setminus \{i\}$ such that $v_i v_j \in E$ and $v_{i+1} v_{j+1} \in E$. (Indeed, the # of $j \in \{1, 2, \ldots, n\} \setminus \{i\}$ that satisfy $v_i v_j \in E$ is $\deg(v_i)$, whereas the # of $j \in \{1, 2, \ldots, n\} \setminus \{i\}$ that satisfy $v_{i+1} v_{j+1} \in E$ is $\deg(v_{i+1})$. The sum of these two #s is $\geq n$, but the total # of $j \in \{1, 2, \ldots, n\} \setminus \{i\}$ is only $n - 1$. Thus, by the pigeonhole principle, there is a j common to both classes.)

Pick such a j, and modify our listing as follows: When it gets to v_i, don't move on to v_{i+1} but instead take the edge $v_i v_j$ to v_j. Then, go backwards to v_{i+1} and jump ahead to v_{j+1} using the edge $v_{i+1} v_{j+1}$. Then continue as before. This modification removes the non-edge $v_i v_{i+1}$ and the possible edge $v_j v_{j+1}$ from our listing, but adds the two definite edges $v_i v_j$ and $v_{i+1} v_{j+1}$. Thus, the hamness of our listing increases by 1 or by 2. This proves Claim 1. □

By induction, this also proves the theorem. □

Corollary 1.14.3 (Dirac)**.** Let $G = (V, E)$ be a simple graph with $n \geq 3$ vertices.

Assume that $\deg x \geq \dfrac{n}{2}$ for each vertex $x \in V$. Then, G has a hamc.

Proof. $\deg x \geq \dfrac{n}{2} \Longrightarrow \deg x + \deg y \geq n$. □

1.14.3. A necessary criterion

Now on to necessary criteria.

Proposition 1.14.4. Let $G = (V, E)$ be a simple graph.

For each subset S of V, we let $G \setminus S$ be the induced subgraph of G on the set $V \setminus S$.

We let $\operatorname{conn}(H)$ denote the number of connected components of a graph H.

(a) If G has a hamc, then every nonempty $S \subseteq V$ satisfies $\operatorname{conn}(G \setminus S) \leq |S|$.

(b) If G has a hamp, then every $S \subseteq V$ satisfies $\operatorname{conn}(G \setminus S) \leq |S| + 1$.

Proof. See the board. □

1.14.4. Hypercubes

Let us now move on to a concrete example of a graph that has a hamc.

Definition 1.14.5. Let $n \in \mathbb{N}$. The ***n*-hypercube** Q_n (more precisely, the ***n*-hypercube graph**) is the simple graph with vertex set

$$\{0,1\}^n = \{(a_1, a_2, \ldots, a_n) \mid \text{each } a_i \text{ belongs to } \{0,1\}\}$$

and edge set defined as follows: A vertex $(a_1, a_2, \ldots, a_n)$ is adjacent to a vertex $(b_1, b_2, \ldots, b_n)$ if and only if there is **exactly one** $i \in \{1, 2, \ldots, n\}$ such that $a_i \neq b_i$. (For example, $(0, 1, 1, 0)$ is adjacent to $(0, 1, 0, 0)$ in Q_4.)

The elements of $\{0,1\}^n$ are called **bitstrings** or **binary words**; their entries are called their **bits** or **letters**. So two bitstrings are adjacent in Q_n if and only if they differ in exactly one bit.

We abbreviate a bitstring $(a_1, a_2, \ldots, a_n)$ as $a_1 a_2 \cdots a_n$.

Theorem 1.14.6 (Gray)**.** Let $n \geq 2$. Then, the graph Q_n has a hamc. (Such hamcs are called **Gray codes**.)

Proof. We will show something stronger:

> *Claim 1:* For each $n \geq 1$, the n-hypercube Q_n has a hamp from $00\cdots0$ to $100\cdots0$. (These are bitstrings, not numbers, obviously.)

Proof of Claim 1. We induct on n.

Base case: Clear from a look at Q_1.

Induction step: Fix $n \geq 1$. Assume that Q_n has a hamp $\mathbf{p}$ from $\underbrace{00\cdots0}_{n \text{ zeroes}}$ to $1\ \underbrace{00\cdots0}_{n-1 \text{ zeroes}}$.

By attaching a 0 to the front of each bitstring in $\mathbf{p}$, we obtain a path $\mathbf{q}$ from $0\underbrace{00\cdots0}_{n \text{ zeroes}}$ to $01\ \underbrace{00\cdots0}_{n-1 \text{ zeroes}}$ in Q_{n+1}.

By attaching a 1 to the front of each bitstring in $\mathbf{p}$, we obtain a path $\mathbf{r}$ from $1\underbrace{00\cdots0}_{n \text{ zeroes}}$ to $11\ \underbrace{00\cdots0}_{n-1 \text{ zeroes}}$ in Q_{n+1}.

Now, we assemble a hamp from $\underbrace{00\cdots0}_{n+1 \text{ zeroes}}$ to $1\underbrace{00\cdots0}_{n \text{ zeroes}}$ in Q_{n+1} as follows: First walk forward along $\mathbf{q}$, then move from $01\ \underbrace{00\cdots0}_{n-1 \text{ zeroes}}$ to $11\ \underbrace{00\cdots0}_{n-1 \text{ zeroes}}$ along the edge that joins these two vertices, and then walk backward along $\mathbf{r}$. This is a hamp because it contains all vertices starting with 0 in its $\mathbf{q}$-part and all vertices starting with 1 in its $\mathbf{r}$-part. So the induction is complete. □

With Claim 1 proved, the theorem follows, since you can get back from $100\cdots0$ to $00\cdots0$ by a single edge. □

1.14.5. Cartesian products

Gray's theorem can in fact be generalized. For this we need the notion of Cartesian products:

Definition 1.14.7. Let $G = (V, E)$ and $H = (W, F)$ be two simple graphs. Their **Cartesian product** $G \times H$ is defined to be the graph $(V \times W,\ E' \cup F')$, where

$$E' := \{(v_1, w)\,(v_2, w) \mid v_1 v_2 \in E \text{ and } w \in W\} \qquad \text{and}$$
$$F' := \{(v, w_1)\,(v, w_2) \mid w_1 w_2 \in F \text{ and } v \in V\}.$$

In other words, it is the graph whose vertices are pairs $(v, w) \in V \times W$ and whose edges are given as follows:

- Two pairs (v_1, w) and (v_2, w) with the same second entry are adjacent if and only if v_1 and v_2 are adjacent in V.
- Two pairs (v, w_1) and (v, w_2) with the same first entry are adjacent if and only if w_1 and w_2 are adjacent in W.
- There are no other adjacencies in $V \times W$.

For example, the Cartesian product $G \times P_2$ of a simple graph G with the 2-path graph P_2 can be constructed by overlaying two copies of G and additionally joining each vertex of the first copy with the corresponding vertex of the second. In particular:

Proposition 1.14.8. We have $Q_n \cong Q_{n-1} \times P_2$ for each $n \geq 1$.

Now we claim the following:

Theorem 1.14.9. Let G and H be two simple graphs. Assume that each of the two graphs G and H has a hamp. Then:

(a) The Cartesian product $G \times H$ has a hamp.

(b) If $|\mathrm{V}(G)|$ or $|\mathrm{V}(H)|$ is even, then $G \times H$ has a hamc.

Proof. See the reference in the notes. □

Now, using the theorem and the proposition, we can reprove Gray's theorem.

1.14.6. Subset graphs

The n-hypercube Q_n can be reinterpreted in terms of subsets of $\{1, 2, \ldots, n\}$. Namely, each bitstring $a_1 a_2 \cdots a_n$ becomes the subset $\{i \in \{1, 2, \ldots, n\} \mid a_i = 1\}$. Thus, we can define a graph

$$G_n = (\mathcal{P}(\{1, 2, \ldots, n\}), E),$$

where $E = \{(U, V) \mid U = V \cup \{v\} \text{ or } V = U \cup \{u\}\}$, and we have $G_n \cong Q_n$. Gray's theorem therefore says that G_n has a hamc.

Now assume that $n \geq 3$ is odd, and consider the induced subgraph of G_n on the set of all subsets of sizes $\dfrac{n \pm 1}{2}$. Alternatively, you can consider the subgraph Q'_n of Q_n defined by

$$Q'_n = \left\{ a_1 a_2 \cdots a_n \in \{0, 1\}^n \mid a_1 + a_2 + \cdots + a_n \in \left\{ \frac{n-1}{2}, \frac{n+1}{2} \right\} \right\}.$$

A recent result of Torsten Mütze (2014) says that Q'_n still has a hamc.

2. Multigraphs

2.1. Definitions

So far, we have been working with simple graphs. We shall now introduce several other kinds of graphs, starting with **multigraphs**:

Definition 2.1.1. Let V be a set. Then, $\mathcal{P}_{1,2}(V)$ shall mean the set of all 1-element or 2-element subsets of V. That is,

$$\mathcal{P}_{1,2}(V) := \{S \subseteq V \mid |S| \in \{1,2\}\} = \{\{u,v\} \mid u,v \in V\}$$

(note that u and v here are not necessarily distinct).

Definition 2.1.2. A **multigraph** is a triple (V, E, φ), where V and E are two finite sets, and $\varphi : E \to \mathcal{P}_{1,2}(V)$ is a map.

Example 2.1.3. See the whiteboard for an example. Formally speaking, this is the multigraph (V, E, φ), where

$$V = \{1,2,3,4,5\}, \qquad E = \{\alpha, \beta, \gamma, \delta, \varepsilon, \kappa, \lambda\},$$

and where $\varphi : E \to \mathcal{P}_{1,2}(V)$ is the map given by

$$\begin{aligned} &\varphi(\alpha) = \{1,2\}, && \varphi(\beta) = \{2,3\}, && \varphi(\gamma) = \{2,3\}, \\ &\varphi(\delta) = \{4,5\}, && \varphi(\varepsilon) = \{4,5\}, && \varphi(\kappa) = \{4,5\}, \\ &\varphi(\lambda) = \{1\}. \end{aligned}$$

This suggests the following terminology (mostly imitating our existing terminology for simple graphs):

Definition 2.1.4. Let $G = (V, E, \varphi)$ be a multigraph. Then:

1. The elements of V are called the **vertices** of G. The set V is called the **vertex set** of G, and is denoted by $\operatorname{V}(G)$.
2. The elements of E are called the **edges** of G. The set E is called the **edge set** of G, and is denoted by $\operatorname{E}(G)$.
3. If e is an edge of G, then the **endpoints** of e are the elements of $\varphi(e)$. (Note the difference to simple graphs: In a multigraph, an edge e does not literally contain its endpoints, but rather gets them assigned to it by the map φ.)
4. We say that an edge e **contains** a vertex v if $v \in \varphi(e)$.

5. Two vertices u and v are said to be **adjacent** if there is an edge of G with endpoints u and v. (This allows $u = v$, if there is an edge e with $\varphi(e) = \{u\}$.)

6. Two edges e and f are said to be **parallel** if $\varphi(e) = \varphi(f)$.

7. We say that G has **no parallel edges** if no two distinct edges are parallel.

8. An edge e is called a **loop** (or **self-loop**) if $\varphi(e)$ is a 1-element set (i.e., if e has only one endpoint).

9. We say that G is **loopless** if G has no loops.

10. The **degree** $\deg v$ (also written $\deg_G v$) of a vertex v of G is defined to be the number of edges that contain v, where loops are counted twice. In other words,

$$\deg v = \deg_G v := \underbrace{|\{e \in E \mid v \in \varphi(e)\}|}_{\substack{\text{this counts all edges} \\ \text{that contain } v}} + \underbrace{|e \in E \mid \varphi(e) = \{v\}|}_{\substack{\text{this counts all loops} \\ \text{that contain } v \text{ again}}}.$$

(Unlike for simple graphs, $\deg v$ is **not** the # of neighbors of v.)

11. A **walk** in G means a list of the form

$$(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k) \qquad \text{with } k \geq 0,$$

where $v_0, v_1, \ldots, v_k$ are vertices of G, where $e_1, e_2, \ldots, e_k$ are edges of G, and where each $i \in \{1, 2, \ldots, k\}$ satisfies

$$\varphi(e_i) = \{v_{i-1}, v_i\}$$

(that is, the endpoints of the edge e_i are v_{i-1} and v_i).

The **vertices** of a walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ are $v_0, v_1, \ldots, v_k$; the **edges** of this walk are $e_1, e_2, \ldots, e_k$; the walk **starts** at v_0 and **ends** at v_k; its **length** is k, its **starting point** is v_0, and its **ending point** is v_k.

12. A **path** means a walk whose vertices are distinct.

13. The notions of "**path-connected**" and "**connected**" and "**component**" are defined exactly as for simple graphs. The symbol $\simeq_G$ still means "path-connected".

14. A **closed walk** (or **circuit**) means a walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ with $v_k = v_0$.

15. A **cycle** means a closed walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ such that

- the vertices $v_0, v_1, \ldots, v_{k-1}$ are distinct;
- the edges $e_1, e_2, \ldots, e_k$ are distinct;
- we have $k \geq 1$.

(Note that we are not requiring $k \geq 3$ any more. Thus, in our example, $(2, \beta, 3, \gamma, 2)$ and $(1, \lambda, 1)$ are cycles, but $(2, \beta, 3, \beta, 2)$ is not.)

16. Hamiltonian paths and cycles are defined as for simple graphs.

17. We draw a multigraph by drawing each vertex as a point, each edge as a curve, and labeling both the vertices and the edges.

So we see two differences between simple graphs and multigraphs:

1. A multigraph can have loops, whereas a simple graph cannot.

2. In a simple graph, an edge e **is** a set of two vertices, whereas in a multigraph, an edge e **has** a set of two vertices (possibly equal, if e is a loop) assigned to it by the map φ. This not only allows for parallel edges, but also lets us store some information in the "identities" of the edges.

Nevertheless, the two notions have a lot in common, so we shall use the word "**graph**" for both of them, and only say "simple graph" or "multigraph" when we want to be specific.

We will next define what properties multigraphs have in common with simple graphs, and what properties they don't. But first, let me discuss conversions between these two concepts.

2.2. Conversions

We can turn each multigraph into a simple graph, at the cost of losing some information:

Definition 2.2.1. Let $G = (V, E, \varphi)$ be a multigraph. Then, the **underlying simple graph** G^{simp} of G means the simple graph

$$(V, \ \{\varphi(e) \ \mid \ e \in E \text{ is not a loop}\}).$$

In other words, it is the simple graph with vertex set V in which two distinct vertices u and v are adjacent if and only if they are adjacent in G. This simple graph is obtained from G by removing loops and "collapsing" parallel edges into a single edge.

This clearly destroys information.

In the inverse direction (simple graph to multigraph), there is an information-preserving conversion:

Definition 2.2.2. Let $G = (V, E)$ be a simple graph. Then, the **corresponding multigraph** G^{mult} of G is defined to be the multigraph

$$(V, E, \iota),$$

where $\iota : E \to \mathcal{P}_{1,2}(V)$ is the map defined by $\iota(e) = e$.

Proposition 2.2.3.

(a) If G is a simple graph, then $\left(G^{\text{mult}}\right)^{\text{simp}} = G$.

(b) If G is a loopless multigraph that has no parallel edges, then $\left(G^{\text{simp}}\right)^{\text{mult}} \cong G$ (just an isomorphism, not an equality, since the "identities" of the edges are lost).

(c) If G is a multigraph that has loops or (distinct) parallel edges, then the multigraph $\left(G^{\text{simp}}\right)^{\text{mult}}$ has fewer edges than G.

Proof. Easy from the definitions. □

We will often identify a simple graph G with the corresponding multigraph G^{mult}. This is occasionally dangerous, since notions are defined somewhat differently for simple graphs and for multigraphs, but usually there is no problem. For instance, the notions of cycle disagree, but can be converted into each other:

Proposition 2.2.4. Let G be a simple graph. Then:

(a) If $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ is a cycle of the multigraph G^{mult}, then $(v_0, v_1, \ldots, v_k)$ is a cycle of the simple graph G.

(b) Conversely, if $(v_0, v_1, \ldots, v_k)$ is a cycle of the simple graph G, then $(v_0, \{v_0, v_1\}, v_1, \{v_1, v_2\}, v_2, \ldots, v_{k-1}, \{v_{k-1}, v_k\}, v_k)$ is a cycle of the multigraph G^{mult}.

Proof. Notes. □

2.3. Generalizing from simple graphs to multigraphs

Let us now go over (most of) our previous theorems about simple graphs and see which of them hold for multigraphs.

2.3.1. Ramsey

Recall the "$R(3, 3) = 6$" theorem from Ramsey theory:

Proposition 2.3.1. Let G be a simple graph with at least 6 vertices (that is, $|\operatorname{V}(G)| \geq 6$). Then, at least one of the following two statements holds:

- **Statement 1:** There exist three distinct vertices a, b, c of G such that ab, bc and ca are edges of G.
- **Statement 2:** There exist three distinct vertices a, b, c of G such that none of ab, bc and ca is an edge of G.

Is this still true if G is a multigraph? Yes, because if G is a multigraph, then we can apply this proposition to G^{simp}, and then it says the exact same thing as it would for G.

2.3.2. Degrees

Recall that we defined a degree in a multigraph in such a way that loops are counted twice. So the degree is not literally the number of edges through a vertex. Also, it is not the number of neighbors any more. Thus, $\deg v \in \{0, 1, \ldots, n-1\}$ (for an n-vertex graph) no longer holds for multigraphs. Nevertheless, some properties of degrees still hold. Most importantly, Euler's equality still holds:

Proposition 2.3.2 (Euler 1736 for multigraphs)**.** Let G be a multigraph. Then, the sum of the degrees of all vertices of G equals twice the number of edges of G. In other words,

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

Proof. See the notes. The main point is that loops are counted twice in the appropriate degree on the LHS, which matches the 2 factor on the RHS. □

As a consequence, the handshake lemma still holds:

Corollary 2.3.3 (handshake lemma)**.** Let G be a multigraph. Then, the number of vertices v of G whose degree $\deg v$ is odd is even.

What about the fact that in a simple graph G with ≥ 2 vertices, there are two distinct vertices with the same degree: Does this still hold for multigraphs? No.

What about Mantel's theorem, saying that each simple graph G with n vertices and more than $n^2/4$ edges has a triangle? Not true for multigraphs, since loops or parallel edges could blow up the number of edges without creating triangles.

2.3.3. Graph isomorphisms

Definition 2.3.4. Let $G = (V, E, \varphi)$ and $H = (W, F, \psi)$ be two multigraphs.

(a) A **graph isomorphism** (or **isomorphism**) from G to H means a **pair** (α, β) of bijections

$$\alpha : V \to W \qquad \text{and} \qquad \beta : E \to F$$

with the property that if $\varphi(e) = \{u, v\}$, then $\psi(\beta(e)) = \{\alpha(u), \alpha(v)\}$. (That is, if an edge e of G goes to an edge f of H under β, then the endpoints of e go to the endpoints of f under α.)

(b) We say that G and H are **isomorphic** (written $G \cong H$) if there exists a graph isomorphism from G to H. Again, this is an equivalence relation.

2.3.4. Cycle graphs

We have previously defined the path graphs P_n for all $n \in \mathbb{N}$ and the cycle graphs C_n for $n > 2$. Now we define C_n for $n = 1$ and $n = 2$:

Definition 2.3.5. We define the 2-nd cycle graph C_2 to be the multigraph with two vertices 1 and 2 and two parallel edges joining these two vertices.

Definition 2.3.6. We define the 1-st cycle graph C_1 to be the multigraph with one vertex 1 and one loop.

Thus, the n-th cycle graph C_n has exactly n edges for each $n \geq 1$ (not just for $n > 2$).

2.3.5. Submultigraphs

Definition 2.3.7. A **submultigraph** of a multigraph $G = (V, E, \varphi)$ is a multigraph of the form (W, F, ψ), where $W \subseteq V$ and $F \subseteq E$ and $\psi = \varphi\mid_F$.

With these definitions, we can now view cycles in a multigraph as subgraphs isomorphic to a cycle graph, including cycles of length 1 as C_1's and cycle of length 2 as C_2's.

Definition 2.3.8. Let $G = (V, E, \varphi)$ be a multigraph. Let S be a subset of V. The **induced submultigraph of** G **on the set** S means the submultigraph

$$\left(S,\ E',\ \varphi\mid_{E'}\right)$$

of G, where

$$E' := \{e \in E \mid \text{all endpoints of } e \text{ belong to } S\}.$$

We sometimes denote this induced submultigraph by $G[S]$.

2.3.6. Disjoint unions

Disjoint unions of multigraphs are defined just as for simple graphs, with the obvious changes.

2.3.7. Walks

We already defined walks, paths, closed walks and cycles for multigraphs. Recall that the length of a walk is still its number of edges. Now let us check up on their basic properties: which of them still hold for multigraphs?

First of all, the edges of a path are still distinct. This still follows just as easily from the definition.

Next, we see that two walks can still be "spliced" together:

Proposition 2.3.9. Let G be a multigraph. Let u, v, w be three vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k)$ be a walk from u to v. Let $\mathbf{b} = (b_0, f_1, b_1, f_2, b_2, \ldots, f_k, b_k)$ be a walk from v to w. Then,

$$\begin{aligned} \mathbf{a} * \mathbf{b} &:= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k, f_1, b_1, f_2, b_2, \ldots, f_k, b_k) \\ &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, b_0, f_1, b_1, f_2, b_2, \ldots, f_k, b_k) \\ &= (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, v, f_1, b_1, f_2, b_2, \ldots, f_k, b_k) \end{aligned}$$

is a walk from u to w.

Walks and paths can still be reversed (= walked backwards):

Proposition 2.3.10. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, e_2, a_2, \ldots, e_k, a_k)$ be a walk from u to v. Then,

$$\operatorname{rev} \mathbf{a} := (a_k, e_k, a_{k-1}, e_{k-1}, \ldots, e_1, a_0)$$

is a walk from v to u, called the **reversal** of $\mathbf{a}$. If $\mathbf{a}$ is a path, then $\operatorname{rev} \mathbf{a}$ is also a path.

Walks that are not paths contain smaller walks between the same vertices:

Proposition 2.3.11. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a}$ be a walk from u to v having length k. Assume that $\mathbf{a}$ is not a path. Then, there exists a walk from u to v whose length is smaller than k.

Corollary 2.3.12 (When there is a walk, there is a path)**.** Let G be a multigraph. Let u and v be two vertices of G. Assume that there is a walk from u to v of length k for some $k \in \mathbb{N}$. Then, there is a path from u to v of length $\leq k$.

2.3.8. Path-connectedness

The theory of path-connectedness and (connected) components looks exactly the same for multigraphs as it does for simple graphs:

Definition 2.3.13. Let G be a multigraph. We define a binary relation $\simeq_G$ on the set $\operatorname{V}(G)$ as follows: For two vertices u and v, we write $u \simeq_G v$ if and only if there exists a walk from u to v in G. By the corollary above, this is equivalent to the existence of a path from u to v in G.

This binary relation $\simeq_G$ is called "**path-connectedness**" or just "**connectedness**". When two vertices u and v satisfy $u \simeq_G v$, we say that u and v are **path-connected**.

Proposition 2.3.14. Let G be a multigraph. Then, the relation $\simeq_G$ is an equivalence relation.

Definition 2.3.15. Let G be a multigraph. The equivalence classes of the path-connectedness relation $\simeq_G$ are called the **connected components** (or, short, **components**) of G.

We say that G is **connected** if G has exactly one component.

Proposition 2.3.16. Let G be a multigraph. Let C be a component of G. Then, the induced subgraph of G on the subset C is connected. (In short: A connected component is connected.)

Proposition 2.3.17. Let G be a multigraph. Let $C_1, C_2, \ldots, C_k$ be all its components (with no repetition). Then,

$$G \cong G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k].$$

2.3.9. Closed walks and cycles

The basic theory of closed walks is similar to that for simple graphs, but the proofs are a bit different (see §3.3.9 in the notes for details):

Definition 2.3.18. A walk $\mathbf{w} = (w_0, e_1, w_1, e_2, w_2, \ldots, e_k, w_k)$ in a multigraph G is called **backtrack-free** if there exists no i such that $e_{i-1} = e_i$. (In other words, if no two consecutive edges of $\mathbf{w}$ are identical.)

Proposition 2.3.19. Let G be a multigraph. Let $\mathbf{w}$ be a backtrack-free walk of G. Then, $\mathbf{w}$ either is a path or contains a cycle.

Corollary 2.3.20. Let G be a multigraph that has a closed backtrack-free walk of length > 0. Then, G has a cycle.

Theorem 2.3.21. Let G be a multigraph. Let u and v be two vertices in G. Assume that there are two distinct backtrack-free walks from u to v. Then, G has a cycle.

2.3.10. The longest path trick?

Recall the following:

Proposition 2.3.22. Let G be a simple graph with at least one vertex. Let $d > 1$ be an integer. Assume that each vertex of G has degree $\geq d$. Then, G has a cycle of length $\geq d + 1$.

This is **not** true for multigraphs! Just imagine a graph with a single vertex and d loops around it. Then, the degree of this vertex is $2d \geq d$, but there are no cycles of length $\geq d + 1$.

2.3.11. Bridges

The concept of removing an edge from a graph works for multigraphs just as it does for simple graphs.

Definition 2.3.23. Let $G = (V, E, \varphi)$ be a multigraph. Let e be an edge of G. Then, $G \setminus e$ will mean the graph obtained from G by removing the edge e, that is,

$$G \setminus e = \left(V,\ E \setminus \{e\},\ \varphi\mid_{E \setminus \{e\}}\right).$$

Some authors denote this graph by $G - e$.

Theorem 2.3.24. Let G be a multigraph. Let e be an edge of G. Then:

- **(a)** If e is an edge of some cycle of G, then the components of $G \setminus e$ are precisely the components of G. (Keep in mind that the components are sets – they are the equivalence classes of the path-connectedness relation.)

- **(b)** If e appears in no cycle of G, then the graph $G \setminus e$ has one more component than G.

The proofs are the same as for simple graphs.

We can define bridges and cut-edges in the same way as for simple graphs, and then of course we conclude that bridges and cut-edges are the same thing, by the theorem.

2.3.12. Dominating sets

Dominating sets can be defined for multigraphs just as for simple graphs: they are sets $S \subseteq V$ of vertices such that every vertex $v \in V \setminus S$ has a neighbor in S. However, this gives you nothing new: The dominating sets of a multigraph G are just the dominating sets of the simple graph G^{simp}.

2.3.13. Hamiltonian paths and cycles

Hamiltonian paths and cycles can be defined for multigraphs just as for simple graphs, but again the question of existence does not really changes: A multigraph G has a Hamiltonian cycle/path if and only if the simple graph G^{simp} has one, with the exception of $|\operatorname{V}(G)| = 1$.

Unfortunately, Dirac's and Ore's theorems do not survive the generalization, since degrees are no longer counting neighbors.

2.4. Eulerian circuits and walks

2.4.1. Definitions

Let us now move on to a new feature of multigraphs, one that we have not studied yet (even for simple graphs).

Recall that a Hamiltonian path/cycle has to use each vertex exactly once. In contrast, a Eulerian walk/circuit has to use each edge exactly once. Here is the precise definition:

Definition 2.4.1. Let G be a multigraph.

(a) A walk of G is said to be **Eulerian** if each edge of G appears exactly once in this walk.

(b) An **Eulerian circuit** of G means a circuit (i.e., closed walk) of G that is Eulerian.

Usually, these are not paths/cycles.

2.4.2. The Euler–Hierholzer theorem

The **Euler–Hierholzer theorem** answers the question about existence of Eulerian walks/circuits for connected multigraphs:

Theorem 2.4.2 (Euler, Hierholzer)**.** Let G be a connected multigraph. Then:

(a) The multigraph G has an Eulerian circuit if and only if each vertex of G has even degree.

(b) The multigraph G has an Eulerian walk if and only if all but at most two vertices of G have even degree.

Euler realized the "$\Longrightarrow$" directions of both parts in his 1736 paper. The proofs are easy: If v is any vertex of G, then an Eulerian circuit must leave and enter v the same number of times, so the degree $\deg v$ must be even. Thus, the "$\Longrightarrow$" direction for part **(a)** is clear. The "$\Longrightarrow$" direction of part **(b)** is not much harder: The same argument yields that $\deg v$ is even for all vertices v except perhaps the starting and the ending points of the Eulerian walk.

The harder part are the "$\Longleftarrow$" directions. We will first focus on part **(a)**. Hierholzer proved it back in 1873; we give a different proof.

We begin with a definition:

Definition 2.4.3. Let G be a multigraph. A **trail** of G means a walk of G whose edges are distinct.

So a trail can repeat vertices, but not edges. Note that an Eulerian walk must always be a trail. A trail cannot be longer than an Eulerian walk. So we can try to construct an Eulerian walk by starting with a trail, and making it longer and longer until it becomes Eulerian. This is precisely the idea of the proof we will do.

We begin with some obvious lemmas:

Lemma 2.4.4. Let G be a multigraph with at least one vertex. Then, G has a longest trail.

Proof. Clearly, G has finitely many trails, and at least one of them (a trivial trail on a single vertex). □

We say that an edge e of a multigraph G **intersects** a walk $\mathbf{w}$ if at least endpoint of e is a vertex of $\mathbf{w}$.

Lemma 2.4.5. Let G be a connected multigraph. Let $\mathbf{w}$ be a walk of G. Assume that there exists an edge of G that is not an edge of $\mathbf{w}$.

Then, there exists an edge of G that is not an edge of $\mathbf{w}$ but intersects $\mathbf{w}$.

Proof. Use the connectedness of G to find a path from an endpoint of the edge to a vertex of $\mathbf{w}$. Then, pick an appropriate edge on this path. □

Lemma 2.4.6. Let G be a multigraph such that each vertex of G has even degree. Let $\mathbf{w}$ be a longest trail of G. Then, $\mathbf{w}$ is a closed walk.

Proof. Assume the contrary. Let u be the starting point and v the ending point of $\mathbf{w}$. Since we assumed that $\mathbf{w}$ is not closed, we have $u \neq v$.

Consider the edges of $\mathbf{w}$ that contain v. Such edges come in two kinds: Those by which $\mathbf{w}$ enters v, and those by which $\mathbf{w}$ leaves v. Except for the very last edge of $\mathbf{w}$, each edge of the former kind is immediately followed by an edge of the latter kind, and vice versa. Thus, the walk $\mathbf{w}$ has exactly one more edge entering v than it has edges leaving v. Hence, the total # of edges of $\mathbf{w}$ that contain v (with loops counting twice) is odd. However, the total # of edges of G that contain v is even (since v has even degree). Thus, the two #s cannot be equal. So there is at least one edge of G that contains v and is not used by $\mathbf{w}$. Thus, we can extend our trail $\mathbf{w}$ by this edge and still have a trail. But this contradicts the fact that $\mathbf{w}$ is a longest trail. Qed. □

Now we are ready to prove the Euler–Hierholzer theorem:

Proof of the Euler–Hierholzer theorem. **(a)** $\Longrightarrow$: Done above.

$\Longleftarrow$: Assume that each vertex of G has even degree.

By our first lemma above, G has a longest trail. Fix such a longest trail and call it $\mathbf{w}$.

By our third lemma above, $\mathbf{w}$ is a closed walk.

We claim that $\mathbf{w}$ is Eulerian. Indeed, assume the contrary. Then, there exists an edge of G that is not an edge of $\mathbf{w}$. Therefore, our second lemma above entails that there exists such an edge that intersects $\mathbf{w}$. Let e be this edge, and let v be a vertex that it has in common with $\mathbf{w}$. Now, we rotate the closed walk $\mathbf{w}$ so that it begins and ends at v, and then extend it by the edge e. We obtain a trail that is longer than $\mathbf{w}$. Contradiction to the definition of $\mathbf{w}$. So the $\Longleftarrow$ direction is proved.

(b) $\Longrightarrow$: Done above.

$\Longleftarrow$: Assume that all but at most two vertices of G have even degree. In other words, at most two vertices of G have odd degree. By the handshake lemma, the # of vertices of G having odd degree is even. So this # is either 0 or 2.

If this # is 0, then part **(a)** yields the existence of an Eulerian circuit, hence an Eulerian walk.

So let us WLOG assume that this # is 2. In other words, G has exactly two distinct vertices u and v with odd degree. Add a new edge e joining u with v to G (even if G already has such an edge!), producing a new multigraph G'. Then, all vertices of G' have even degree (since the degrees of u and v have increased by 1, while all other degrees are unchanged from G). Thus, by part **(a)**, this new graph G' has an Eulerian circuit. Now, cutting this circuit at the new edge e, we obtain an Eulerian walk of G. This proves the $\Longleftarrow$ direction. □

The above proof is algorithmic, although I've done my best to hide it.

3. Digraphs and multidigraphs

3.1. Definition

So far we have encountered two concepts of graphs: simple graphs and multigraphs. They have one thing in common: the two endpoints of an edge are equal in rights. In other words, an edge is a "two-way road". Such graphs are useful for modelling symmetric relations.

Now we shall introduce two analogous concepts for non-symmetric relations, where the edges have directions. These are known as **directed graphs**, short **digraphs**. In such directed graphs, each edge will have a specified starting point (its "source") and a specified ending point (its "target"). Correspondingly we draw these edges as arrows, and only allow them to be used in the forward direction in a walk. Here are the definitions in detail:

Definition 3.1.1. A **simple digraph** is a pair (V, A), where V is a finite set, and where A is a subset of $V \times V$.

Definition 3.1.2. Let $D = (V, A)$ be a simple digraph.

(a) The set V is called the **vertex set** of D; it is denoted by $\operatorname{V}(D)$.

Its elements are called the **vertices** (or the **nodes**) of D.

(b) The set A is called the **arc set** of D; it is denoted by $\operatorname{A}(D)$.

Its elements are called the **arcs** (or **directed edges**) of D.

We will occasionally abbreviate a pair (u, v) as uv.

(c) If (u, v) is an arc of D (or, more generally, a pair in $V \times V$), then u is called the **source** of this arc, and v is called the **target** of this arc.

(d) We draw D as follows: Each vertex of D is depicted as a point, and each arc uv is depicted as an arrow from the u-point to the v-point.

Note that simple digraphs (unlike simple graphs) are allowed to have loops (i.e., arcs of the form vv).

Definition 3.1.3. A **multidigraph** is a triple (V, A, ψ), where V and A are two finite sets, and $\psi : A \to V \times V$ is a map.

Definition 3.1.4. Let $D = (V, A, \psi)$ be a multidigraph.

(a) The set V is called the **vertex set** of D; it is denoted by $\operatorname{V}(D)$.

Its elements are called the **vertices** (or the **nodes**) of D.

(b) The set A is called the **arc set** of D; it is denoted by $\operatorname{A}(D)$.

Its elements are called the **arcs** (or **directed edges**) of D.

(c) If a is an arc of D, and if $\psi(a) = (u, v)$, then the vertex u is called the **source** of a, whereas the vertex v is called the **target** of a.

(d) We draw D as follows: Each vertex of D is depicted as a point, and each arc a is depicted as an arrow from the u-point to the v-point, where $(u, v) = \psi(a)$.

Definition 3.1.5. The word "**digraph**" (short for "**directed graph**") means either "simple digraph" or "multidigraph" depending on the context.

3.2. Outdegrees and indegrees

Definition 3.2.1. Let D be a digraph with vertex set V and arc set A. Let $v \in V$ be any vertex. Then:

(a) The **outdegree** of v denotes the number of arcs of D whose source is v. This outdegree is denoted by $\deg^+ v$.

(b) The **indegree** of v denotes the number of arcs of D whose target is v. This indegree is denoted by $\deg^- v$.

Recall that in a graph, the sum of all degrees is twice the number of edges. Here is an analogue for digraphs:

Proposition 3.2.2 (diEuler)**.** Let D be a digraph with vertex set V and arc set A. Then,

$$\sum_{v \in V} \deg^+ v = \sum_{v \in V} \deg^- v = |A| .$$

Proof. By the definition of outdegree, we have

$$\begin{aligned}\sum_{v \in V} \deg^+ v &= \sum_{v \in V} (\text{the number of arcs of } D \text{ whose source is } v) \\ &= (\text{the number of all arcs of } D) = |A| .\end{aligned}$$

Similarly for indegrees. □

3.3. Conversions

3.3.1. Multidigraphs to multigraphs

Any multidigraph D can be turned into an (undirected) graph G by "removing the arrowheads" (aka "forgetting the directions of the arcs"):

Definition 3.3.1. Let D be a multigraph. Then, D^{und} will denote the multigraph obtained from D by replacing each arc with an edge whose endpoints are the source and the target of this arc. Formally: If $D = (V, A, \psi)$, then $D^{\text{und}} = (V, A, \varphi)$, where $\varphi : A \to \mathcal{P}_{1,2}(V)$ is the map sending each arc a to the set of entries of $\psi(a)$ (that is, if $\psi(a) = (u, v)$, then $\varphi(a) := \{u, v\}$).

Note that the arcs are being reused as the edges; only the map ψ is replaced.

3.3.2. Multigraphs to multidigraphs

Conversely, we can turn a multigraph G into a multidigraph G^{bidir} by "duplicating" each edge (more precisely: turning each edge into two arcs with opposite orientations). Formally:

Definition 3.3.2. Let $G = (V, E, \varphi)$ be a multigraph. For each edge $e \in E$, let us choose one of the endpoints of e and call it s_e; the other endpoint will be called t_e. (If e is a loop, then $t_e = s_e$.)

We then define G^{bidir} to be the multidigraph $(V,\ E \times \{1, 2\},\ \psi)$, where the map $\psi : E \times \{1, 2\} \to V \times V$ is defined by the rules

$$\psi(e, 1) = (s_e, t_e) \qquad \text{and}$$
$$\psi(e, 2) = (t_e, s_e).$$

We call G^{bidir} the **bidirectionalized multidigraph** of G.

Note that the map ψ depends on our choice of s_e's. But different choices lead to isomorphic multidigraphs. (The notion of **isomorphism** of multidigraphs is exactly the one you expect.)

The operation $G \mapsto G^{\text{bidir}}$ is injective, whereas the operation $D \mapsto D^{\text{und}}$ is not. Note, however, that $\left(G^{\text{bidir}}\right)^{\text{und}}$ is not isomorphic to G, since each edge of G is doubled in $\left(G^{\text{bidir}}\right)^{\text{und}}$.

3.3.3. Simple digraphs to multidigraphs

The next operation we define converts simple digraphs to multidigraphs, just as we did for simple graphs:

Definition 3.3.3. Let $D = (V, A)$ be a simple digraph. Then, the **corresponding multidigraph** D^{mult} is defined to be the multidigraph

$$(V, A, \iota),$$

where $\iota : A \to V \times V$ is the map sending each $a \in A$ to a itself.

3.3.4. Multidigraphs to simple digraphs

The final conversion is an operation $D \mapsto D^{\text{simp}}$ that turns multidigraphs into simple digraphs:

Definition 3.3.4. Let $D = (V, A, \psi)$ be a multidigraph. Then, the **underlying simple digraph** D^{simp} of D is the simple digraph

$$(V, \{\psi(a) \mid a \in A\}).$$

In other words, this is the simple digraph with vertex set V in which there is an arc from u to v if there exists one in D. In other words, you "collapse" parallel arcs into a single arc.

3.3.5. Multidigraphs as a big tent

We now have constructed several conversions between our four kinds of graphs (simple graphs, multigraphs, simple digraphs and multidigraphs). Note that each kind can be converted injectively (i.e., without forgetting/losing any information) into multidigraphs:

- Each simple graph becomes a multigraph via $G \mapsto G^{\text{mult}}$.
- Each multigraph becomes a multidigraph via $G \mapsto G^{\text{bidir}}$.
- Each simple digraph becomes a multidigraph via $D \mapsto D^{\text{mult}}$.

So multidigraphs are the most general of the four notions. In particular, any theorem about them can be specialized to any of the other types.

3.4. Subdigraphs

Subdigraphs and **induced subdigraphs** of digraphs are defined just as in the undirected case (see the notes).

3.5. Walks, paths, closed walks, cycles

Let us now define various kinds of walks for simple digraphs and for multidigraphs.

For simple digraphs, we imitate the definition from simple graphs, just making sure to require that all arcs are used in the correct direction:

Definition 3.5.1. Let D be a simple digraph. Then:

(a) A **walk** (in D) means a finite sequence $(v_0, v_1, \ldots, v_k)$ of vertices of D (with $k \geq 0$) such that all of the pairs $v_0v_1,\ v_1v_2,\ \ldots,\ v_{k-1}v_k$ are arcs of D. (The latter condition is vacuously true if $k = 0$.)

(b) If $\mathbf{w} = (v_0, v_1, \ldots, v_k)$ is a walk in D, then:

a) the **vertices** of **w** are defined to be $v_0, v_1, \ldots, v_k$;

b) the **arcs** of **w** are the arcs $v_0v_1,\ v_1v_2,\ \ldots,\ v_{k-1}v_k$;

c) the **length** of **w** is k (that is, the # of arcs);

d) the **starting point** of **w** is v_0, and we say that **w starts** at v_0;

e) the **ending point** of **w** is v_k, and we say that **w ends** at v_k.

(c) A **path** (in D) means a walk whose vertices are distinct.

(d) A **walk/path from** p **to** q means a walk/path with starting point p and ending point q.

(e) A **closed walk** (aka **circuit**) means a walk $(v_0, v_1, \ldots, v_k)$ with $v_0 = v_k$.

(f) A **cycle** of D means a closed walk $(v_0, v_1, \ldots, v_k)$ such that $k \geq 1$ and such that the vertices $v_0, v_1, \ldots, v_{k-1}$ are distinct.

Note that cycles require $k \geq 1$, not $k \geq 3$ as for simple graphs.

Now let us define the same concepts for multidigraphs:

Definition 3.5.2. Let $D = (V, A, \psi)$ be a multidigraph. Then:

(a) A **walk** (in D) means a list of the form

$$(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k) \qquad (\text{with } k \geq 0),$$

where $v_0, v_1, \ldots, v_k$ are vertices of D, and where $a_1, a_2, \ldots, a_k$ are arcs of D, and where

$$\psi(a_i) = (v_{i-1}, v_i) \qquad \text{for each } i \in \{1, 2, \ldots, k\}$$

(that is, each arc a_i has source v_{i-1} and target v_i).

(b) If $\mathbf{w} = (v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ is a walk in D, then:

a) the **vertices** of **w** are defined to be $v_0, v_1, \ldots, v_k$;

b) the **arcs** of **w** are the arcs $a_1, a_2, \ldots, a_k$;

c) the **length** of **w** is k (that is, the # of arcs);

d) the **starting point** of **w** is v_0, and we say that **w starts** at v_0;

e) the **ending point** of **w** is v_k, and we say that **w ends** at v_k.

(c) A **path** (in D) means a walk whose vertices are distinct.

(d) A **walk/path from** p **to** q means a walk/path with starting point p and ending point q.

(e) A **closed walk** (aka **circuit**) means a walk $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ with $v_0 = v_k$.

(f) A **cycle** of D means a closed walk $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ such that $k \geq 1$ and such that the vertices $v_0, v_1, \ldots, v_{k-1}$ are distinct. (This automatically implies that the arcs $a_1, a_2, \ldots, a_k$ are distinct, since each arc a_i has source v_{i-1}.)

Now let us see which properties of walks etc. still hold for digraphs:

- Splicing walks still works: If **a** is a walk from u to v and **b** is a walk from v to w, then $\mathbf{a} * \mathbf{b}$ is a walk from u to w.
- Reversing walks does not work any more.
- If there is a walk from u to v, then there is a path from u to v, whose length is at most the length of the walk.
- Any walk is either a path or contains a cycle.

3.6. Connectivity

We defined the "path-connected" relation for undirected graphs using the existence of paths or walks. For a digraph, however, the relations "there is a walk from u to v" and "there is a walk from v to u" are not the same, so the same definition would no longer produce an equivalence relation. So I prefer not to give this relation a label like $\simeq_D$. Instead, we define **strong path-connectedness** to mean the existence of **both** walks:

Definition 3.6.1. Let D be a multidigraph. We define a binary relation $\simeq_D$ on the set $\operatorname{V}(D)$ as follows: For two vertices u and v of D, we shall have $u \simeq_D v$ if and only if there exists a walk from u to v and there exists a walk from v to u.

This binary relation $\simeq_D$ is called **strong path-connectedness**.

Proposition 3.6.2. This relation $\simeq_D$ is an equivalence relation, and it does not change if we replace the word "walk" by "path" both times.

Proof. Just as in the undirected case. □

Definition 3.6.3. Let D be a multidigraph. The equivalence classes of the equivalence relation $\simeq_D$ are called the **strong components** of D.

We say that D is **strongly connected** if D has exactly one strong component.

Thus, a multidigraph D is strongly connected if and only if it has at least one vertex, and there is a path from any vertex to any vertex.

There is also a weaker notion of components and connectedness:

Definition 3.6.4. Let D be a multidigraph. Then, the **weak components** of D are defined to be the components of the undirected graph D^{und}.

We say that D is **weakly connected** if D has exactly one weak component (i.e., if D^{und} is connected).

Proposition 3.6.5. Any strongly connected digraph is weakly connected.

Proof. Any walk of D is (or, to be precise, gives rise to) a walk of D^{und}. □

Let us take a quick look at what bidirectionalization (i.e., the operation $G \mapsto G^{\text{bidir}}$) does to walks, paths, closed walks and cycles:

Proposition 3.6.6. Let G be a multigraph. Then:

- **(a)** The walks of G are "more or less the same as" the walks of G^{bidir}. More precisely, each walk of G gives rise to a walk of G^{bidir}, and conversely, each walk of G^{bidir} gives rise to a walk of G. These are one-to-one correspondences if G is loopless.
- **(b)** The paths of G are "more or less the same as" the path of G^{bidir}. This time, this is always a one-to-one correspondence.
- **(c)** The closed walks of G are "more or less the same as" the closed walks of G^{bidir}.
- **(e)** The cycles of G are not quite the same as the cycles of G^{bidir}. Indeed, if e is an edge of G with endpoints u and v, then (u, e, v, e, u) is not a cycle of G, but either $(u, (e,1), v, (e,2), u)$ or $(u, (e,2), v, (e,1), u)$ is a cycle of G^{bidir}.

3.7. Eulerian walks and circuits

Now let us extend the notions of Eulerian walks and circuits to multidigraphs.

Definition 3.7.1. Let D be a multidigraph.

(a) A walk of D is said to be **Eulerian** if each arc of D appears exactly once in this walk.

(b) An **Eulerian circuit** means a circuit that is Eulerian.

Now we have the following analogue of the Euler–Hierholzer theorem (proof on the homework):

Theorem 3.7.2 (diEuler, diHierholzer)**.** Let D be a weakly connected multidigraph. Then:

(a) The multidigraph D has an Eulerian circuit if and only if each vertex v satisfies $\deg^+ v = \deg^- v$.

(b) The multidigraph D has an Eulerian walk if and only if all but two vertices v satisfy $\deg^+ v = \deg^- v$, whereas the remaining two vertices v satisfy $\left|\deg^+ v - \deg^- v\right| \leq 1$.

Definition 3.7.3. A multidigraph D is said to be **balanced** if each vertex v of D satisfies $\deg^+ v = \deg^- v$.

Proposition 3.7.4. Let G be a multigraph. Then, the multidigraph G^{bidir} is balanced.

Combining this easy proposition with the above theorem, we obtain the following:

Theorem 3.7.5. Let G be a connected multigraph. Then, the multidigraph G^{bidir} has an Eulerian circuit. In other words, there is a circuit of G that contains each edge **exactly twice**, and uses it once in each direction.

3.8. Hamiltonian cycles and paths

We can define **Hamiltonian paths** (**hamps**) and **Hamiltonian cycles** (**hamcs**) for simple digraphs just as we did for simple graphs: they are paths/cycles that contain each vertex exactly once (where for a cycle, of course, $v_0 = v_k$ count as one time).

Ore's theorem has a directed analogue (which is much harder to prove):

Theorem 3.8.1 (Meyniel)**.** Let $D = (V, A)$ be a strongly connected loopless simple digraph with n vertices. Assume that for each pair $(u, v) \in V \times V$ of two distinct vertices u and v that are not adjacent in either direction (i.e., that satisfy $(u, v) \notin A$ and $(v, u) \notin A$), we have $\deg u + \deg v \geq 2n - 1$, where $\deg w := \deg^+ w + \deg^- w$. Then, D has a hamc.

There are some more interesting facts about hamps and hamcs in simple digraphs, but more on them next time.

3.9. The reverse and complete digraphs

Definition 3.9.1. Let $D = (V, A)$ be a simple digraph. Then:

(a) The elements of $(V \times V) \setminus A$ will be called the **non-arcs** of D.

(b) The **reversal** of a pair $(i, j) \in V \times V$ means the pair (j, i).

(c) We define D^{rev} as the simple digraph (V, A^{rev}), where

$$A^{\text{rev}} = \{(j, i) \mid (i, j) \in A\}.$$

This is the digraph obtained from D by reversing each arc (i.e., swapping source and target). This is called the **reversal** of D.

(d) We define $\overline{D}$ as the simple digraph $(V, (V \times V) \setminus A)$. This is called the **complement** of D. It is obtained from D by removing all the arcs and adding all the non-arcs as arcs.

Proposition 3.9.2 (obvious)**.** Let D be a simple digraph. Then,

$$(\text{\# of hamps of } D^{\text{rev}}) = (\text{\# of hamps of } D).$$

Theorem 3.9.3 (Berge)**.** Let D be a simple digraph. Then,

$$\left(\text{\# of hamps of } \overline{D}\right) \equiv (\text{\# of hamps of } D) \bmod 2.$$

For a proof, see the notes or the video.

3.10. Tournaments

We now introduce a special class of digraphs:

Definition 3.10.1. A **tournament** is defined to be a loopless simple digraph D that satisfies the

- **Tournament axiom:** For any two distinct vertices u and v of D, **exactly** one of (u, v) and (v, u) is an arc of D.

("**Loopless**" means that D has no loops.)

A tournament can be viewed as a complete graph, whose each edge has been given a direction. In particular, for any n-element set V, there are exactly $2^{n(n-1)/2}$ many tournaments with vertex set V.

Theorem 3.10.2 (Easy Rédei theorem)**.** A tournament always has at least one hamp (if you allow the empty list to be a hamp for the empty tournament).

Theorem 3.10.3 (Hard Rédei theorem)**.** Let D be a tournament. Then,

$$(\text{\# of hamps of } D) \text{ is odd.}$$

Proof. See the notes. □

Remark 3.10.4. What can the # of hamps of a tournament be? Can it be just any odd positive integer?

Surprisingly, 7 and 21 are impossible. Besides these, no other impossible numbers are known. All odd numbers between 1 and 80555 are possible except for 7 and 21. See MathOverflow #232751 for details.

3.11. Hamiltonian cycles in tournaments

Every tournament has a hamp, but not every tournament has a hamc. One obstruction is clear:

Proposition 3.11.1. If a digraph D has a hamc, then D is strongly connected.

In general, this is only a necessary, not a sufficient requirement for the existence of a hamc. However, for tournaments, it is also sufficient, as long as the tournament has at least two vertices:

Theorem 3.11.2 (Camion's theorem)**.** If a tournament D is strongly connected and has at least two vertices, then D has a hamc.

Proof. Notes. □

3.12. Application of tournaments to the Vandermonde determinant

A curious application of tournaments is a combinatorial proof of the Vandermonde determinant formula (see the notes for details). Recall this formula:

Theorem 3.12.1 (Vandermonde determinant formula)**.** Let $x_1, x_2, \ldots, x_n$ be n numbers (or, more generally, elements of a commutative ring). Consider the $n \times n$-matrix

$$V := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \cdots & x_n^{n-1} \end{pmatrix} = \left(x_j^{i-1}\right)_{i,j\in\{1,2,\ldots,n\}}.$$

Then, its determinant is

$$\det V = \prod_{1\le i<j\le n} \left(x_j - x_i\right).$$

There are many nice proofs of this. The following proof goes back to Ira Gessel 1979:

First of all, how do $\det V$ and $\prod\limits_{1\le i<j\le n} \left(x_j - x_i\right)$ relate to tournaments?

As a warmup, assume that some number $y_{(i,j)}$ is given for each pair (i,j) of integers. Let's expand the product

$$\left(y_{(1,2)} + y_{(2,1)}\right)\left(y_{(1,3)} + y_{(3,1)}\right)\left(y_{(2,3)} + y_{(3,2)}\right).$$

We get a sum of 8 addends, each of which is a product of one of $y_{(1,2)}, y_{(2,1)}$ with one of $y_{(1,3)}, y_{(3,1)}$ and one of $y_{(2,3)}, y_{(3,2)}$. This can be expressed as

$$\sum_{\substack{D \text{ is a tournament} \\ \text{with vertex set } \{1,2,3\}}} \ \prod_{(i,j) \text{ is an arc of } D} y_{(i,j)}.$$

More generally, for each n, we have

$$\prod_{1\le i<j\le n} \left(y_{(i,j)} + y_{(j,i)}\right) = \sum_{\substack{D \text{ is a tournament} \\ \text{with vertex set } \{1,2,\ldots,n\}}} \ \prod_{(i,j) \text{ is an arc of } D} y_{(i,j)}.$$

Substituting

$$y_{(i,j)} = \begin{cases} x_j, & \text{if } i<j; \\ -x_j, & \text{if } i>j \end{cases}$$

in this equality, we obtain

$$\prod_{1\leq i<j\leq n}\left(x_j - x_i\right) = \sum_{\substack{D \text{ is a tournament} \\ \text{with vertex set } \{1,2,\ldots,n\}}} \prod_{(i,j) \text{ is an arc of } D} \begin{cases} x_j, & \text{if } i<j; \\ -x_j, & \text{if } i>j \end{cases}$$
$$= \sum_{\substack{D \text{ is a tournament} \\ \text{with vertex set } \{1,2,\ldots,n\}}} (-1)^{(\text{\# of red arcs of } D)} \prod_{j=1}^{n} x_j^{\deg^- j},$$

where an arc (i,j) is called "red" if $i>j$. We call this the "big sum".

On the other hand, if we let S_n be the group of permutations of $\{1,2,\ldots,n\}$, and if we denote the sign of a permutation $\sigma \in S_n$ by $\operatorname{sign}\sigma$, then

$$\det V = \det\left(V^T\right) = \det\left(x_i^{j-1}\right)_{i,j\in\{1,2,\ldots,n\}}$$
$$= \sum_{\sigma\in S_n} \operatorname{sign}\sigma \cdot \prod_{j=1}^{n} x_j^{\sigma(j)-1}.$$

We call this the "small sum".

Our goal is to prove that this small sum equals the big sum above. To prove this, we must verify the following:

1. Each addend of the small sum is an addend of the big sum. Namely, for each $\sigma \in S_n$, we find an appropriate tournament T_σ that has

$$(-1)^{(\text{\# of red arcs of } T_\sigma)} \prod_{j=1}^{n} x_j^{\deg^- j} = \operatorname{sign}\sigma \cdot \prod_{j=1}^{n} x_j^{\sigma(j)-1}.$$

2. All the addends of the big sum that are **not** addends of the small sum cancel each other out. Why?

   The idea is to argue that if a tournament D appears in the big sum but not in the small sum, then D has a 3-cycle (i.e., a cycle of length 3). When we reverse such a 3-cycle (i.e., reverse each arc of it), the indegrees of all vertices are preserved, but the sign $(-1)^{(\text{\# of red arcs of } D)}$ flips.

   This suffices to show that for each addend that appears in the big but not in the small sum, there is another addend with the same magnitude but opposite sign. Unfortunately, this by itself is not sufficient to ensure that all these addends cancel out; for example, the sum $1+1+1+(-1)$ has the same property but is not 0.

   What we need is to ensure that for each given magnitude, there are equally many positive and negative addends of this magnitude.

   One way to obtain this is to define an actual bijection, by deciding **which** 3-cycle to reverse.

Another way is to focus on the tournaments with exactly k many 3-cycles (for a given $k > 0$) and a given sequence of indegrees. Then, you show that **among these tournaments**, there are equally many that have an odd # of red arcs as there are that have an even # of red arcs. Thus the sub-sum ranging over these particular tournaments yields 0. This turns out to come from the directed version of the handshake lemma!

As mentioned above, see the lecture notes for more (and the reference therein for details).

3.13. The adjacency matrix

Definition 3.13.1. Let $D = (V, A, \psi)$ be a multidigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

Let C be the $n \times n$-matrix (with real entries) defined by

$$C_{i,j} = (\text{the number of all arcs } a \in A \text{ with source } i \text{ and target } j) \qquad \text{for all } i, j \in V.$$

(The notation $M_{i,j}$ means the (i, j)-th entry of a matrix M.)

This matrix C is called the **adjacency matrix** of D.

This matrix determines D up to the identities of the arcs.

Theorem 3.13.2. Let $D = (V, A, \psi)$ be a multidigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

Let C be the adjacency matrix of D.

Then, for any $k \in \mathbb{N}$ and any $i, j \in V$, we have

$$(\text{the number of all walks from } i \text{ to } j \text{ having length } k) = \left(C^k\right)_{i,j}.$$

Proof. See §4.5.4 in the notes. (Idea: $C^k = CC^{k-1}$.) □

4. Trees and arborescences

Trees are particularly nice graphs. They can be characterized as

- the connected multigraphs that have no cycles;

- the minimal connected graphs on a given set of vertices;
- the maximal acyclic (= having no cycles) graphs on a given set of vertices;
- in many other ways.

Arborescences are their closest analogue for directed graphs.

In this chapter, we will discuss the theory of trees and some of their applications.

4.1. Some properties of connected components

Definition 4.1.1. Let G be a multigraph. Then, $\operatorname{conn} G$ means the nubmer of components of G.

So a multigraph G is connected if and only if $\operatorname{conn} G = 1$. Moreover, it has no vertices if and only if $\operatorname{conn} G = 0$.

Recall that if G is a multigraph and e is an edge of G, then:

- If e belongs to a cycle of G, then the components of $G \setminus e$ are just the components of G.
- If e belongs to no cycle of G, then $G \setminus e$ has one more component than G.

Therefore:

Corollary 4.1.2 (1)**.** Let G be a multigraph. Let e be an edge of G. Then:

(a) If e belongs to a cycle of G, then $\operatorname{conn}(G \setminus e) = \operatorname{conn} G$.

(b) If e belongs to no cycle of G, then $\operatorname{conn}(G \setminus e) = \operatorname{conn} G + 1$.

(c) In either case, $\operatorname{conn}(G \setminus e) \leq \operatorname{conn} G + 1$.

Corollary 4.1.3 (2)**.** Let $G = (V, E, \varphi)$ be a multigraph. Then, $\operatorname{conn} G \geq |V| - |E|$.

Proof. Start with the graph $(V, \varnothing, \varnothing)$, which has all the vertices of G but none of the edges. This graph has $|V|$ components. Now add in the edges $e \in E$ one by one. Each time you add an edge, the # of components goes down by at most 1 (by part **(c)** of corollary 1). Thus, at the end, you have at least $|V| - |E|$ components.

(See the notes for a formalization of this argument by induction on $|E|$.) □

Corollary 4.1.4 (3)**.** Let $G = (V, E, \varphi)$ be a multigraph that has no cycles. Then, $\operatorname{conn} G = |V| - |E|$.

Proof. Argue as in the proof of corollary 2, but use part **(b)** of corollary 1 rather than part **(c)** because your new edges never end up in a cycle. □

Corollary 4.1.5 (4)**.** Let $G = (V, E, \varphi)$ be a multigraph that has at least one cycle. Then, $\operatorname{conn} G \geq |V| - |E| + 1$.

Proof. Fix an edge $e \in E$ that belongs to a cycle. Then, $\operatorname{conn} G = \operatorname{conn}(G \setminus e)$ by part **(a)**. But corollary 2 yields $\operatorname{conn}(G \setminus e) \geq |V| - |E \setminus \{e\}| = |V| - |E| + 1$. Combine. □

Let us summarize:

Theorem 4.1.6. Let $G = (V, E, \varphi)$ be a multigraph. Then:

(a) We always have $\operatorname{conn} G \geq |V| - |E|$.

(b) We have $\operatorname{conn} G = |V| - |E|$ if and only if G has no cycles.

Remark 4.1.7. Does the difference $\operatorname{conn} G - (|V| - |E|)$ count the cycles of G ? Sadly, no. Nevertheless, it is a useful quantity, known as the **cyclomatic number** of G.

4.2. Forests and trees

Definition 4.2.1. A **forest** is a multigraph with no cycles.
(In particular, it cannot have loops or parallel edges.)

Definition 4.2.2. A **tree** is a connected forest.

Trees can be described in many ways:

Theorem 4.2.3 (Tree equivalence theorem)**.** Let $G = (V, E, \varphi)$ be a multigraph. Then, the following eight statements are equivalent:

- **Statement T1:** The multigraph G is a tree.
- **Statement T2:** The multigraph G has no loops, and we have $V \neq \varnothing$, and for each $u, v \in V$, there is a **unique** path from u to v.
- **Statement T3:** We have $V \neq \varnothing$, and for each $u, v \in V$, there is a **unique** backtrack-free walk from u to v.

- **Statement T4:** The multigraph G is connected, and we have $|E| = |V| - 1$.
- **Statement T5:** The multigraph G is connected, and we have $|E| < |V|$.
- **Statement T6:** We have $V \neq \varnothing$, and the graph G is a forest, but adding any new edge to G creates a cycle.
- **Statement T7:** The multigraph G is connected, but removing any edge from G yields a disconnected (= non-connected) graph.
- **Statement T8:** The multigraph G is a forest, and we have $|E| \geq |V| - 1$ and $V \neq \varnothing$.

Proof. To prove T1$\Longrightarrow$T4, we observe that if G is a tree, then $\operatorname{conn} G = 1$, but also G has no cycles, so part **(b)** of the theorem above yields that $\operatorname{conn} G = |V| - |E|$, thus $|V| - |E| = \operatorname{conn} G = 1$ and thus $|E| = |V| - 1$. So T1$\Longrightarrow$T4.

This argument can also be reversed, yielding T4$\Longrightarrow$T1.

The implication T4$\Longrightarrow$T5 is obvious.

Let us prove T5$\Longrightarrow$T8: Assume T5, so that G is connected and $|E| < |V|$. We want to show that $|E| \geq |V| - 1$ and that G is a forest. Since G is connected, we have $\operatorname{conn} G = 1$. If G had any cycles, then the previous theorem would yield $\operatorname{conn} G > |V| - |E| \geq 1$ (since $|E| < |V|$), contradicting $\operatorname{conn} G = 1$. Hence, G has no cycles, i.e., is a forest. The equality case of the previous theorem thus yields $\operatorname{conn} G = |V| - |E|$, so that $|V| - |E| = \operatorname{conn} G = 1$. Hence, $|E| = |V| - 1$, so in particular $|E| \geq |V| - 1$. This proves T8.

And also T4 along the way. So we get T5$\Longrightarrow$T4.

Let us prove T8$\Longrightarrow$T1: Assume T8, so that G is a forest and $|E| \geq |V| - 1$. Since G has no cycles (being a forest), we can use the equality case of the previous theorem, and obtain $\operatorname{conn} G = |V| - |E| \leq 1$ (since $|E| \geq |V| - 1$). Since $\operatorname{conn} G \geq 1$ (because $V \neq \varnothing$), this entails $\operatorname{conn} G = 1$, so that G is connected. Thus, G is a tree, i.e., T1 holds.

The implication T3$\Longrightarrow$T2 is easy: If there is a backtrack-free walk, then there is a path; if the backtrack-free walk is unique, then so is the path. Moreover, looplesnsness follows from the uniqueness of backtrack-free walks (since (v) and (v, e, v) would be different backtrack-free walks).

To prove T1$\Longrightarrow$T3, assume that G is a tree. Then, G has no cycles. Hence, backtrack-free walks are unique, since we previously proved that if there exist two distinct backtrack-free walks from u to v, then there is a cycle. Since G is connected, these walks also exist. So T3 follows.

To prove T2$\Longrightarrow$T1, observe that the existence of backtrack-free walks ensures that G is connected (since $V \neq \varnothing$), whereas their uniqueness (and the looplessness of G) ensures that G has no cycles.

Next, we argue that T7$\Longrightarrow$T1, we observe that if G had a cycle, then we could pick any edge e of this cycle and remove it from G without disconnecting the graph.

To prove T2$\Longrightarrow$T7, we assume T2. In particular, this yields that G is connected. Now, if there was an edge e that we could delete without disconnecting G, then the resulting graph $G \setminus e$ would still have a path between the endpoints of e. Thus, G would have two different paths between these two endpoints: one just being the direct path through e, while the other is that path of $G \setminus e$. This would contradict T2. So T7 follows.

To prove T6$\Longrightarrow$T1, we assume T6. Thus, G is a forest. If G was not connected, then G has at least two components U and U' (since $V \neq \varnothing$). Picking a vertex in U and a vertex in U' and joining them by an edge, we obtain a larger graph but still no cycles (since this newly added edge is the only connection between U and U'), which contradicts T6. Thus, G is connected, and T1 follows.

Remains to prove T1$\Longrightarrow$T6. Assume T1, so G is a tree. If we add a new edge e to G, then this edge – combined with the already existing path between its endpoints in G – creates a cycle. So T6 is true.

Now we have proved enough implications to put all the eight statements T1, T2, ..., T8 in a strongly connected digraph (arcs = implications). So these statements are all equivalent.

(See the notes for a slightly different variant of this proof.) □

Proposition 4.2.4. Let G be a multigraph, and let $C_1, C_2, \ldots, C_k$ be its components. Then, G is a forest if and only if all the induced subgraphs $G[C_1], G[C_2], \ldots, G[C_k]$ are trees.

Proof. See the notes. □

Warning 4.2.5. Computer scientists use the word "tree" for a similar but different object: they have a chosen vertex called "root", and often the neighbors of a vertex come with a chosen ordering.

4.3. Leaves

Definition 4.3.1. Let T be a tree. A vertex of T is called a **leaf** if its degree is 1.

A tree with $n \geq 3$ vertices cannot have more than $n-1$ leaves (since otherwise, each vertex would be a leaf, but this would result in a disconnected graph). The number $n-1$ is indeed achieved by the simple graph

$$(\{0,1,\ldots,n-1\}, \{0i \mid i > 0\}),$$

which is a tree and is called an n-**star graph**.

A tree with $n \geq 2$ vertices cannot have fewer than 2 leaves (to be proved in a moment). The number 2 is indeed achieved by the path graph

$$P_n = \quad \circ - \circ - \circ - \circ \cdots \circ - \circ\,.$$

Theorem 4.3.2. Let T be a tree with at least 2 vertices. Then:

(a) The tree T has at least 2 leaves.

(b) Let v be a vertex of T. Then, v lies on a path between two distinct leaves of T. (That is, there exist two distinct leaves p and q of T such that v lies on the path from p to q.)

Proof. **(b)** Among all paths that contain v, let $\mathbf{w}$ be a longest one. Let p and q be its starting point and its ending point. If p was not a leaf, then there would be an edge of T that contains p but is not an edge of $\mathbf{w}$. Thus, we could either attach it to $\mathbf{w}$ and obtain a longer path (contradiction) or we could attach it to a piece of $\mathbf{w}$ and obtain a cycle (also contradiction because trees have no cycles). Therefore, we get a contradiction in either case, and conclude that p is a leaf. Similarly, q is a leaf. Moreover, v is on the path from p to q, because $\mathbf{w}$ is a path from p to q and contains v.

If $p = q$, then $\mathbf{w}$ is the trivial path (v), but this is absurd, since the longest path containing v is certainly not the trivial path (after all, there is at least one edge containing v). So p and q are distinct, and we are done.

(a) Follows from **(b)**. □

Leaves are particularly helpful for induction proofs, because of the following theorem:

Theorem 4.3.3 (induction principle for trees)**.** Let T be a tree with at least 2 vertices. Let v be a leaf of T. Let $T \setminus v$ be the multigraph obtained from T by removing v and all edges containing v (there is only one such edge). Then, $T \setminus v$ is again a tree.

Proof. The graph $T \setminus v$ is a subgraph of T, and so is a forest (since T is a forest).

It remains to show that $T \setminus v$ is connected. But v is a leaf, so has only one edge through it. Thus, a path between two vertices p and q of T cannot contain v unless v is one of p and q. Hence, no such path will be destroyed if we remove v from T. As a consequence, $T \setminus v$ remains connected. □

The converse of this theorem is also true:

Theorem 4.3.4. Let G be a multigraph. Let v be a vertex of G such that $\deg v = 1$ and such that $G \setminus v$ is a tree. Then, G is a tree.

Proof. LTTR. □

4.4. Spanning trees

First we define a notion that makes sense for any multigraphs:

Definition 4.4.1. A **spanning subgraph** of a multigraph $G = (V, E, \varphi)$ means a multigraph of the form $(V, F, \varphi \mid_F)$, where F is a subset of E.

In other words, it means a submultigraph of G with the same vertex set as G.

So it is a multigraph obtained from G by removing some edges, but leaving all vertices undisturbed.

Definition 4.4.2. A **spanning tree** of a multigraph G means a spanning subgraph of G that is a tree.

A spanning tree of a graph G can be regarded as a minimum "backbone" of G – that is, a way to keep G connected using as few edges as possible. If G is not connected, then this is not possible at all, so G has no spanning trees in this case. The best one can hope for is a spanning subgraph that keeps each component of G connected with a few edges as possible. This is known as a "spanning forest":

Definition 4.4.3. A **spanning forest** of a multigraph G means a spanning subgraph H of G that is a forest and satisfies $\operatorname{conn} H = \operatorname{conn} G$.

When G is connected, a spanning forest is the same as a spanning tree.

The following theorem is crucial:

Theorem 4.4.4. Each connected multigraph G has at least one spanning tree.

First proof. Construct a spanning tree step by step by removing edges that belong to cycles. In fact, if we remove an edge that belongs to a cycle from a connected graph, then the graph remains connected. This cannot go on forever, so eventually it brings us to a graph that is still connected but has no cycles any more – i.e., to a tree. This is our spanning tree.

Note that we slowly transformed G by keeping it connected but gradually removing its cycles. □

Second proof. This is the opposite strategy to the first proof: We start with a totally disconnected graph H with the same vertex set as G (but no edges), and we slowly transform it, keeping it a forest but gradually making it connected.

At each step, we add some edge of G to our subgraph H. Namely, we add an edge e of G to H that is not already in H and joins two different components of H. Each time we do this, the resulting graph still is a forest, because if it had a cycle, then this cycle would be using the newly added edge e, but then this edge e would not join two different components of the previous graph. □

Third proof. The following algorithm is known as "breadth-first search":

We choose an arbitrary vertex r of G, and then progressively "spread a rumor" from r. The rumor starts at r. On day 0, only r has heard it. Every day, every vertex that knows the rumor spreads it to all its neighbors. Since G is connected, the rumor eventually spreads to each vertex. Now, each vertex v (other than r) remembers which other vertex v' it has first heard the rumor from (if there are several, it just chooses one at random). Let us denote by e_v an edge between v and v' (if there are several, choose one). Now, these edges e_v constitute a spanning tree of G.

For the proof of this, see the notes. □

I omit a fourth proof ("depth-first search"), which is in the notes.

4.4.1. Applications

Spanning trees have lots of applications:

- A spanning tree of a graph can be viewed as a kind of "backbone" of the graph, which provides "canonical" paths between any two vertices.
- Minimum spanning trees (each edge of G has a cost, and you are looking for a spanning tree with minimum total cost) solve a gloval version of the cheapest-path problem.
- Depth-first search can be used to navigate mazes.

Here is a more theoretical application of spanning trees:

Definition 4.4.5. A vertex v of a connected multigraph G is called a **cut-vertex** if the graph $G \setminus v$ is disconnected. Otherwise, we call it a **non-cut-vertex**.

Proposition 4.4.6. Let G be a connected multigraph with ≥ 2 vertices. Then, there are at least 2 non-cut-vertices of G.

Proof. Pick a spanning tree T of G. Then, T has at least 2 leaves (as we showed above). Each of these leaves is a non-cut-vertex of G. □

4.4.2. Existence and construction of a spanning forest

Corollary 4.4.7. Each multigraph has a spanning forest.

Proof. Each component of this multigraph is connected, and thus has a spanning tree. Combine these trees into a spanning forest. □

4.5. Centers of graphs and trees

4.5.1. Distances

Given a graph, we can define a "distance" between any two of its vertices, simply by counting the edges on the shortest path from one to the other:

Definition 4.5.1. Let G be a multigraph. Let u and v be two vertices of G.

We define the **distance** between u and v to be the smallest length of a path from u to v. If there is no such path, then we define it to be ∞.

We denote this distance by $d(u,v)$ or $d_G(u,v)$.

Remark 4.5.2. Distances in a multigraph satisfy the rules you would expect:

1. We have $d(u,u) = 0$ for each vertex u.
2. We have $d(u,v) = d(v,u)$ for any vertices u and v.
3. The **triangle inequality**: We have $d(u,v) + d(v,w) \geq d(u,w)$ for any vertices u,v,w.
4. The distances do not change if we replace "path" by "walk".
5. If our multigraph has n vertices, then $d(u,v) \leq n-1$ for any two vertices u and v.

Also, the distances on G are the same as the distance on G^{simp}.

Note that in a tree, the distance between two vertices u and v is just the length of **the** path from u to v.

4.5.2. Eccentricity and centers

Definition 4.5.3. Let v be a vertex of a multigraph $G = (V, E, \varphi)$. The **eccentricity** of v (in G) is defined to be the number

$$\operatorname{ecc}_G v := \max \{d(v,u) \mid u \in V\} \in \mathbb{N} \cup \{\infty\}.$$

Definition 4.5.4. Let $G = (V, E, \varphi)$ be a multigraph. A **center** of G means a vertex of G whose eccentricity is minimum.

4.5.3. The centers of a tree

Theorem 4.5.5. Let T be a tree. Then:

1. The tree T has either 1 or 2 centers.

2. If T has 2 centers, then these 2 centers are adjacent.

3. Moreover, these centers can be found by the following algoirhtm:

   If T has more than 2 vertices, then we remove all leaves from T (simultaneously). What remains is again a tree. If that tree still has more than 2 vertices, then we repeat. And so on until we are left with a tree that has only 1 or 2 vertices. Its vertices are the centers of T.

For the proof, see the notes.

4.6. Arborescences

4.6.1. Definitions

Enough about undirected graph.

What would be a directed version of a tree? Trees are graphs that are connected and have no cycles. This suggests two directed versions:

1. We can study digraphs that are strongly connected and have no cycles. Unfortunately, the only such digraphs have exactly 1 vertex and no arcs.

2. We can study digraphs that are weakly connected and have no cycles. But there are too many.

Generally, digraphs that have no cycles are called **acyclic**, and often are called **dags** (short for "directed acyclic graphs"). But these dags are not quire like trees.

Here is a more convincing analogue of trees for digraphs:

Definition 4.6.1. Let D be a multidigraph. Let r be a vertex of D.

(a) We say that r is a **from-root** (or, short, **root**) of D if for each vertex v of D, the digraph D has a path from r to v.

(b) We say that r is a **to-root** of D if for each vertex v of D, the digraph D has a path from v to r.

(c) We say that D is an **arborescence rooted from** r if r is a from-root of D and the undirected multigraph D^{und} has no cycles.

(d) We say that D is an **arborescence rooted to** r if r is a to-root of D and the undirected multigraph D^{und} has no cycles.

Clearly, the notions of "from-root" and "to-root" are analogues, and in fact get transformed into each other if you reverse each arc of D, and the same applies to "arborescences rooted from r" and "arborescences rooted to r".

4.6.2. Arborescences vs. trees: statement

Theorem 4.6.2. Let D be a multidigraph, and let r be a vertex of D. Then, the following two statements are equivalent:

- **Statement C1:** The multidigraph D is an arborescence rooted from r.
- **Statement C2:** The undirected multigraph D^{und} is a tree, and each arc of D is "oriented away from r" (this means the following: the source of this arc lies on the unique path between r and the target of this arc on D^{und}).

Proof. Easy to see on the whiteboard. Or read it up in the notes. □

There is an analogous result about arborescences rooted to r.

4.6.3. The arborescence equivalence theorem

Theorem 4.6.3 (The arborescence equivalence theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph with a from-root r. Then, the following six statements are equivalent:

- **Statement A1:** The multidigraph D is an arborescence rooted from r.
- **Statement A2:** We have $|A| = |V| - 1$.
- **Statement A3:** The multidigraph D^{und} is a tree.
- **Statement A4:** For each vertex $v \in V$, the multidigraph D has a unique walk from r to v.
- **Statement A5:** If we remove any arc from D, then r will no longer be a from-root.
- **Statement A6:** We have $\deg^- r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$.

Proof. Whiteboard. (We proved A1$\Longrightarrow$A3$\Longrightarrow$A2$\Longrightarrow$A1 and A3$\Longrightarrow$A4$\Longrightarrow$A6$\Longleftrightarrow$A5$\Longrightarrow$A2.) (Alternatively, see the notes.) □

4.7. Arborescences vs. trees

The above theorems suggest that an arborescence is "essentially" a tree with a chosen root. The orientations of the arcs are then dictated by which vertex is closer to the root. We can make this precise:

Definition 4.7.1. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Let e be an edge of T. It is easy to see that the distances from the two endpoints of e to the vertex r differ by exactly 1. So one of them is smaller than the other.

1. We define the ***r*-parent** of e to be the endpoint of e whose distance to r is the smallest. We denote it e^{-r}.

2. We define the ***r*-child** of e to be the endpoint of e whose distance to r is the largest. We denote it e^{+r}.

So $d(r, e^{+r}) = d(r, e^{-r}) + 1$.

Definition 4.7.2. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Then, we define a multidigraph $T^{r\rightarrow}$ to be

$$T^{r\rightarrow} := (V, E, \psi),$$

where

$$\psi(e) := \left(e^{-r}, e^{+r}\right) \qquad \text{for each } e \in E.$$

Colloquially: $T^{r\rightarrow}$ is the multidigraph obtained from T by "orienting" each edge e "away" from the root r (meaning that it becomes an arc from e^{-r} to e^{+r}).

Theorem 4.7.3. Let D be a multidigraph, and let r be a vertex of D. Then, the following two statements are equivalent:

- **Statement C1:** The multidigraph D is an arborescence rooted from r.

- **Statement C2:** The undirected multigraph D^{und} is a tree, and $D = \left(D^{\text{und}}\right)^{r\rightarrow}$ (this is an equality, not an isomorphism!).

We stated this already; see §5.7 in the notes for a detailed proof.

Proposition 4.7.4. Let D be an arborescence rooted from r. Then, r is the **only** from-root of D.

Proof. Statement A6 in the arborescence equivalence theorem entails that $\deg^- r = 0$. So r cannot be reached from any other vertex. Thus, no other vertex can be a from-root. □

Definition 4.7.5. A multidigraph D is said to be an **arborescence** if there exists a vertex r of D such that D is an arborescence rooted from r. In this case, r is unique, as we just saw.

Theorem 4.7.6. There are two mutually inverse maps

$$\{\text{pairs } (T,r) \text{ of a tree } T \text{ and a vertex } r \text{ of } T\} \to \{\text{arborescences}\}, \quad (T,r) \mapsto T^{r\to}$$

and

$$\{\text{arborescences}\} \to \{\text{pairs } (T,r) \text{ of a tree } T \text{ and a vertex } r \text{ of } T\}, \quad D \mapsto \left(D^{\text{und}}, \sqrt{D}\right),$$

where $\sqrt{D}$ denotes the root of D.

4.8. Spanning arborescences

In analogy to spanning subgraphs of a multigraph, we can define spanning subdigraphs of a multidigraph:

Definition 4.8.1. A **spanning subdigraph** of a multidigraph $D = (V, A, \psi)$ means a multidigraph of the form $(V, B, \psi\mid_B)$, where B is a subset of A.

In other words, it means a submultidigraph of D with the same vertex set as D.

So it is a multidigraph obtained from D by removing some arcs, but leaving all vertices undisturbed.

Definition 4.8.2. Let D be a multidigraph. Let r be a vertex of D. A **spanning arborescence of** D **rooted from** r means a spanning subdigraph of D that is an arborescence rooted from r.

Theorem 4.8.3. Let D be a multidigraph. Let r be a from-root of D. Then, D has a spanning arborescence rooted from r.

Proof. This is an analogue of "every connected graph has a spanning tree".

We can prove it using breadth-first search just as for the undirected case.

Or we can prove it by removing useless arcs just as for the undirected case.

(I don't know whether the other two methods for constructing spanning trees can be adapted to this case as well.) □

Example 4.8.4. The n-cycle graph C_n has n spanning trees.

The **n-cycle digraph**

$$\overrightarrow{C}_n = \left(\{1,2,\ldots,n\},\ \{12,\ 23,\ \ldots,\ (n-1)\,n,\ n1\}\right)$$

has exactly 1 spanning arborescence rooted from 1.

4.9. The BEST theorem: statement

Here comes something much more surprising.

Recall that a multidigraph $D = (V, A, \psi)$ is **balanced** if and only if each vertex v satisfies $\deg^- v = \deg^+ v$. This is necessary for the existence of an Eulerian circuit. If D is weakly connected, then this is also sufficient (by the directed Euler–Hierholzer theorem).

Unexpectedly, there is a formula for the number of these Eulerian circuits:

Theorem 4.9.1 (The BEST theorem)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph such that each vertex has indegree > 0. Fix an arc a of D, and let r be its target. Let $\tau(D, r)$ be the number of spanning arborescences of D rooted from r. Let $\varepsilon(D, a)$ be the number of Eulerian circuits of D whose last arc is a. Then,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^- u - 1\right)!.$$

The "BEST" in the name is short for de Bruijn, van Aardenne–Ehrenfest, Smith and Tutte, who discovered the theorem in the 1950s.

Note that the $\varepsilon(D, a)$ on the LHS does not actually depend on a. Any Eulerian circuit can be rotated so that it ends with a, and each rotation-equivalence class of Eulerian circuits contains exactly one that ends with a. So $\varepsilon(D, a) = \varepsilon(D, b)$ for all arcs a, b.

So the theorem yields that $\tau(D, r)$ does not depend on r either! We will do this in more details later on.

To prove the BEST theorem, I will restate it for the sake of convenience. Namely, I will flip the direction of each arc, thus replacing arborescences rooted from r by arborescences rooted to r. The main advantage of this is pedagogical, but it is a nice exercise in restating theorems.

4.10. Arborescences rooted to r

Theorem 4.10.1 (The dual arborescence equivalence theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph with a to-root r. Then, the following six statements are equivalent:

- **Statement A'1:** The multidigraph D is an arborescence rooted to r.
- **Statement A'2:** We have $|A| = |V| - 1$.
- **Statement A'3:** The multidigraph D^{und} is a tree.
- **Statement A'4:** For each vertex $v \in V$, the multidigraph D has a unique walk from v to r.
- **Statement A'5:** If we remove any arc from D, then r will no longer be a to-root.
- **Statement A'6:** We have $\deg^+ r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^+ v = 1$.

Proof. This is just the arborescence equivalence theorem, after you reverse each arc. (Details see notes.) □

Let us restate the BEST theorem by reversing all the arcs:

Theorem 4.10.2 (The BEST' theorem)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph such that each vertex has outdegree > 0. Fix an arc a of D, and let r be its source. Let $\tau(D, r)$ be the number of spanning arborescences of D rooted to r. Let $\varepsilon(D, a)$ be the number of Eulerian circuits of D whose first arc is a. Then,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.$$

4.11. The BEST theorem: proof

We shall prove the BEST' theorem, from which the BEST theorem will follow.

Proof of the BEST' theorem. An **outgoing/incoming arc from/into** a vertex v means an arc whose source/target is v.

An a**-Eulerian circuit** means an Eulerian circuit whose first arc is a.

A **sparb** means a spanning arborescence of D rooted to r.

A spanning subdigraph of D always has the form $(V, B, \psi\mid_B)$ for some subset $B \subseteq A$. So we shall identify such a digraph with the corresponding B. Conversely, any subset $B \subseteq A$ becomes a spanning subdigraph of D. So when we say "B is a sparb", we shall mean "$(V, B, \psi\mid_B)$ is a sparb".

For each a-Eulerian circuit $\mathbf{e}$, we define a subset $\operatorname{Exit} \mathbf{e}$ of A as follows:

Clearly, $\mathbf{e}$ begins and ends at r. As an Eulerian circuit, it contains each arc of D and thus each vertex of D (since all outdegrees are > 0). For each vertex $u \in V \setminus \{r\}$, we let $e(u)$ be the **last exit** of $\mathbf{e}$ from u; this means the last arc of $\mathbf{e}$ that has source u. We let $\operatorname{Exit}\mathbf{e}$ be the set of these last exits $e(u)$ for all $u \in V \setminus \{r\}$.

Now I claim:

Claim 1: Let $\mathbf{e}$ be an a-Eulerian circuit. Then, $\operatorname{Exit}\mathbf{e}$ is a sparb.

Claim 2: For each sparb B (regarded as a subset of A), there are exactly $\prod_{u \in V} (\deg^+ u - 1)!$ many a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit}\mathbf{e} = B$.

Once both of these claims are proved, it will follow that the map

$$\{a\text{-Eulerian circuit}\} \to \{\text{sparbs}\}, \qquad \mathbf{e} \mapsto \operatorname{Exit}\mathbf{e}$$

is a $\prod_{u \in V} (\deg^+ u - 1)!$-to-1 correspondence. Thus it will follow that

$$(\#\text{ of } a\text{-Eulerian circuits}) = (\#\text{ of sparbs}) \cdot \prod_{u \in V} (\deg^+ u - 1)!,$$

that is,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} (\deg^+ u - 1)!.$$

So it remains to prove Claim 1 and Claim 2.

Proof of Claim 1. We must show that $\operatorname{Exit}\mathbf{e}$ is a sparb, i.e., a spanning arborescence rooted to r.

First, we show that r is a to-root of $\operatorname{Exit}\mathbf{e}$. Indeed, if we start at any vertex u of D and keep following only the last exit arcs, then we will eventually have to reach r, because the last exit arcs we take are successively later along $\mathbf{e}$ and we eventually run out of arcs.

So $\operatorname{Exit}\mathbf{e}$ has to-root r. Moreover, $\operatorname{Exit}\mathbf{e}$ has $|V| - 1$ arcs, so it satisfies Statement A'2 of the dual arborescence equivalence theorem. Thus, $\operatorname{Exit}\mathbf{e}$ is a sparb. □

Proof of Claim 2. Let $B \subseteq A$ be a sparb. We must prove that there are exactly $\prod_{u \in V} (\deg^+ u - 1)!$ many a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit}\mathbf{e} = B$.

We shall refer to the arcs in B as the B**-arcs**. Since B is a sparb, we know that there is no B-arc with source r; however, for each vertex $u \in V \setminus \{r\}$, there is exactly one B-arc with source u.

Now we are trying to count the a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit}\mathbf{e} = B$.

Let us construct such an a-Eulerian circuit $\mathbf{e}$ as follows:

A turtle wants to walk through the digraph D using each arc of D at most once. It starts its walk by heading out from the vertex r along the arc a. From that point on, it proceeds in the usual way you walk on a digraph (using arcs from source to target), but observing two rules:

1. It never re-uses an arc.

2. It never uses a B-arc until it has exhausted all other outgoing arcs from the current vertex.

Clearly, the turtle will eventually get stuck somewhere. Let $\mathbf{w}$ be the total walk that the turtle has traced by the time it got stuck. Thus, $\mathbf{w}$ is a trail (= a walk with distinct arcs). I claim that $\mathbf{w}$ is an a-Eulerian circuit with $\operatorname{Exit} \mathbf{w} = B$. Why?

- First, we claim that $\mathbf{w}$ is a closed walk (i.e., ends at r).

  [*Proof:* If not, then it ends at a different vertex s, and so it enters s more often than it leaves s. But since D is balanced, this means that at least one outgoing arc from s is still unused.]

- We say that a vertex u of D is **exhausted** if all outgoing arcs from u have been used (i.e., are arcs of $\mathbf{w}$).

  Now we want to show that each vertex of D is exhausted.

  Clearly, r is exhausted (since the turtle got stuck at r).

  Now, we shall show that if the target of a B-arc is exhausted, then so is the source of this B-arc.

  [*Proof:* Let these source and target be called s and t. So there is a B-arc from s to t, and we know that t is exhausted. Why is s exhausted? Well, since t is exhausted, the turtle has used all outgoing arcs from t, and thus also used all incoming arcs into t because D is balanced. Thus, in particular, the B-arc from s to t has been used. But the turtle only uses a B-arc if it has no other options. So s must have been exhausted.]

  Thus, exhaustedness spreads from r to all the sources of the B-arcs with target r, then to all the sources of the B-arcs whose targets are the previous sources, and so on. Since r is a to-root of B (because B is a sparb), this will cover all the vertices of D. Thus, each vertex of D is exhausted. This means that all arcs of D have been used. So $\mathbf{e}$ is an Eulerian circuit.

- By the definition of $\mathbf{e}$, we have $\operatorname{Exit} \mathbf{e} = B$.

So we have shown that $\mathbf{e}$ is an a-Eulerian circuit with $\operatorname{Exit} \mathbf{e} = B$.

The above construction can be made in $\prod_{u \in V} \left(\deg^+ u - 1\right)!$ different ways, because we get to choose, for each vertex $u \in V$, the order in which the turtle uses the outgoing arcs from u, subject to the conditions that

1. for $u = r$, the arc a has to be used first;
2. for $u \neq r$, the B-arc has to be used last.

So Claim 2 is proved.] □

Thus the theorem follows. □

4.12. A corollary about spanning arborescences

Corollary 4.12.1. Let $D = (V, A, \psi)$ be a balanced multidigraph. For each vertex $r \in V$, let $\tau(D, r)$ be the number of spanning arborescences of D rooted to r. Then, $\tau(D, r)$ does not depend on r.

Proof. Obvious if $|V| \leq 1$. Also obvious if some vertex has outdegree 0. Thus, we WLOG assume that $|V| > 1$ and that all outdegrees are > 0. Then, the BEST' theorem yields

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!,$$

where a is any arc with source r. But the LHS does not depend on a or on r, since an Eulerian circuit can be made to start with any arc. So the RHS does not depend on a or on r either. Therefore, $\tau(D, r)$ does not depend on r. □

4.13. Spanning arborescences vs. spanning trees

Let us try to apply the BEST theorem to a digraph of the form G^{bidir}, where G is a multigraph. We need to find the # of spanning arborescences of G^{bidir} rooted to a given vertex r. I claim that this number is just the number of spanning trees of G:

Proposition 4.13.1. Let $G = (V, E, \varphi)$ be a multigraph. Fix a vertex $r \in V$. Recall that the arcs of G^{bidir} are the pairs $(e, i) \in E \times \{1, 2\}$. Identify each spanning tree of G with its edge set, and each spanning arborescence of G^{bidir} with its arc set.

If B is a spanning arborescence of G^{bidir} rooted to r, then we set

$$\overline{B} := \{e \mid (e, i) \in B\}.$$

Then, there is a bijection

$$\left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} \to \{\text{spanning trees of } G\},$$
$$B \mapsto \overline{B}.$$

Proof. Think about it (or look it up in the notes). □

4.14. The matrix-tree theorem

4.14.1. Introduction

By the above proposition, to count spanning trees of a multigraph is a particular case of counting spanning arborescences of a multidigraph. But how can we do either? Let us begin with some simple examples:

n	1	2	3	4	5
# of spanning trees of K_n	1	1	3	16	125

.

This pattern extends: The # of spanning trees of K_n is n^{n-2}.

We will prove this later. For now, however, let us address the more general problem of counting spanning arborescences of an arbitrary digraph D.

4.14.2. Notations

Definition 4.14.1. We will use the **Iverson bracket notation**: If $\mathcal{A}$ is any statement, then

$$[\mathcal{A}] := \begin{cases} 1, & \text{if } \mathcal{A} \text{ is true;} \\ 0, & \text{if } \mathcal{A} \text{ is false.} \end{cases}$$

For example, $[K_2 \text{ is a tree}] = 1$ but $[K_3 \text{ is a tree}] = 0$.

Definition 4.14.2. Let M be a matrix. Let i and j be two integers. Then,

$M_{i,j}$ will mean the entry of M in row i and column j;

$M_{\sim i,\sim j}$ will mean the matrix M without row i and column j.

For example,

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}_{2,3} = f \qquad \text{and} \qquad \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}_{\sim 2,\sim 3} = \begin{pmatrix} a & b \\ g & h \end{pmatrix}.$$

Recall also that the symbol # means "number".

4.14.3. The Laplacian of a multidigraph

We shall now assign a matrix to (more or less) any multidigraph:

Definition 4.14.3. Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

For any $i, j \in V$, we let $a_{i,j}$ be the # of arcs of D having source i and target j.

The **Laplacian** of D is defined to be the $n \times n$-matrix $L \in \mathbb{Z}^{n \times n}$ whose entries are given by

$$L_{i,j} = \left(\deg^+ i\right) \cdot \underbrace{[i = j]}_{\text{aka } \delta_{i,j}} - a_{i,j} \qquad \text{for all } i, j \in V.$$

In other words, it is the matrix

$$L = \begin{pmatrix} \deg^+ 1 - a_{1,1} & -a_{1,2} & \cdots & -a_{1,n} \\ -a_{2,1} & \deg^+ 2 - a_{2,2} & \cdots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \cdots & \deg^+ n - a_{n,n} \end{pmatrix}.$$

Note that loops do not matter for the matrix L. (Indeed, each loop from i to i adds 1 to $\deg^+ i$ but also adds 1 to $a_{i,i}$, and these cancel out in the difference.)

We notice one first property of Laplacians:

Proposition 4.14.4. Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Then, the Laplacian L of D is singular, i.e., we have $\det L = 0$.

Proof. The sum of all columns of L is the zero vector, because for each $i \in V$ we have

$$\begin{aligned} \sum_{j=1}^{n} L_{i,j} &= \sum_{j=1}^{n} \left(\left(\deg^+ i\right) \cdot [i = j] - a_{i,j} \right) \\ &= \underbrace{\sum_{j=1}^{n} \left(\deg^+ i\right) \cdot [i = j]}_{= \deg^+ i} - \underbrace{\sum_{j=1}^{n} a_{i,j}}_{= \deg^+ i} = \deg^+ i - \deg^+ i = 0. \end{aligned}$$

In other words, we have $Le = 0$ for the vector $e := (1, 1, \ldots, 1)^T$. Thus, the vector e lies in the kernel (= nullspace) of L, and so L is singular.

(Note that we need $n > 0$ here, because for $n = 0$, the vector e itself is 0.) □

4.14.4. The Matrix-Tree Theorem: statement

The last proposition shows that the determinant of the Laplacian of a digraph is not very interesting. However, when the a matrix has determinant 0, its submatrices might still have useful determinants, particularly the rank-sized ones (i.e., the largest submatrices whose determinants are nonzero). In the case of the Laplacian, these turn out to count spanning arborescences:

Theorem 4.14.5 (Matrix-Tree Theorem, or short MTT)**.** Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of D. Let r be any vertex of D. Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det\left(L_{\sim r, \sim r}\right).$$

Some remarks:

- The determinant $\det\left(L_{\sim r, \sim r}\right)$ is the (r, r)-th entry of the adjugate matrix of D.
- The $V = \{1, 2, \ldots, n\}$ assumption is a typical "WLOG assumption": If you have an arbitrary digraph D, you can always rename its vertices to ensure that $V = \{1, 2, \ldots, n\}$. Thus, the MTT lets you count the spanning arborescences of any digraph.

4.14.5. Application: Counting the spanning trees of K_n

Before we prove the MTT, let us see how it can be used.

We fix a positive integer n. Let L be the Laplacian of the multidigraph K_n^{bidir} (where K_n is the complete graph on $\{1, 2, \ldots, n\}$ as usual). Each vertex of K_n^{bidir} has outdegree $n-1$, and thus we have

$$L = \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{n \times n}$$

(this is the $n \times n$-matrix whose diagonal entries are $n-1$ and whose off-diagonal entries are -1). By the last proposition from the previous section, there is a bijection

$$\left\{\text{spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right\} \to \left\{\text{spanning trees of } K_n\right\},$$

and therefore

$$\begin{aligned}
&(\text{\# of spanning trees of } K_n) \\
&= \left(\text{\# of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right) \\
&= \det\left(L_{\sim 1,\sim 1}\right) \qquad\qquad (\text{by the MTT}) \\
&= \det \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{(n-1)\times(n-1)} .
\end{aligned}$$

Now how do we compute this determinant? Here are three ways (these require $n \geq 2$; the $n = 1$ case is obvious):

- The most elementary approach is using row transformations:

$$\begin{aligned}
&\det \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{(n-1)\times(n-1)} \\
&= \det \begin{pmatrix} n-1 & -1 & -1 & -1 & \cdots & -1 \\ -n & n & 0 & 0 & \cdots & 0 \\ -n & 0 & n & 0 & \cdots & 0 \\ -n & 0 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -n & 0 & 0 & 0 & \cdots & n \end{pmatrix}_{(n-1)\times(n-1)} \\
&= n^{n-2} \det \begin{pmatrix} n-1 & -1 & -1 & -1 & \cdots & -1 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{(n-1)\times(n-1)} \\
&= n^{n-2} \underbrace{\det \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{(n-1)\times(n-1)}}_{\substack{=1 \\ \text{(since this is a triangular matrix} \\ \text{whose diagonal entries are 1)}}} \\
&= n^{n-2}.
\end{aligned}$$

- The so-called **matrix determinant lemma** says that for any $m \times m$-matrix $A \in \mathbb{R}^{m \times m}$, any column vector $u \in \mathbb{R}^{m \times 1}$ and any row vector $v \in \mathbb{R}^{1 \times m}$, we have
$$\det(A + uv) = \det A + v(\operatorname{adj} A)\, u.$$
This helps us compute our determinant, since
$$\begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}_{(n-1)\times(n-1)} = \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}.$$

- Here is an approach that is heavier on linear algebra (eigenvectors and eigenvalues):
Let $(e_1, e_2, \ldots, e_{n-1})$ be the standard basis of the $\mathbb{R}$-vector space $\mathbb{R}^{n-1}$. Then, we can find the following $n-1$ eigenvectors of our matrix $\begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}$:
  - the eigenvector $e_1 + e_2 + \cdots + e_{n-1} = (1, 1, \ldots, 1)^T$ with eigenvalue 1;
  - the $n-2$ eigenvectors $e_1 - e_i$ for all $i \in \{2, 3, \ldots, n-1\}$ with eigenvalue n.

  Since these $n-1$ eigenvectors are linearly independent, they form a basis of $\mathbb{R}^{n-1}$. Hence, our matrix is similar to the diagonal matrix with diagonal entries $1, \underbrace{n, n, \ldots, n}_{n-2 \text{ times}}$, and thus has determinant $1 \cdot \underbrace{nn \cdots n}_{n-2 \text{ times}} = n^{n-2}$.

There are many other ways. Either way, we obtain the result n^{n-2}. Thus, taking the MTT for granted, we have proved:

Theorem 4.14.6 (Cayley's formula)**.** Let n be a positive integer. Then, the # of spanning trees of the complete graph K_n is n^{n-2}.

Corollary 4.14.7. Let n be a positive integer. Then, the # of simple graphs with vertex set $\{1, 2, \ldots, n\}$ that are trees is n^{n-2}.

There are many ways to prove Cayley's formula. For bijective proofs, see the references in the notes.

Remark 4.14.8. We have counted trees on vertex set $\{1, 2, \ldots, n\}$ – that is, labelled trees. To count unlabelled trees (i.e., isomorphism classes of trees) is much harder, and there is no explicit formula.

4.14.6. Preparations for the proof

Our proof of the MTT is a multi-step process. We begin with an innocent lemma (yet another criterion for a digraph to be an arborescence):

Lemma 4.14.9. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume

- that D has no cycles;
- that D has no arcs with source r;
- that each vertex $v \in V \setminus \{r\}$ has outdegree 1.

Then, D is an arborescence rooted to r.

Proof. It suffices to show that r is a to-root of D, because then the dual arborescence equivalence theorem will yield the claim (since two of our conditions say $\deg^+ r = 0$ and $\deg^+ v = 1$ for each $v \in V \setminus \{r\}$).

Why is r a to-root of D ? We must show that for each vertex v, there is a walk from v to r. Start walking from v, always taking the unique outgoing arc. This walk will eventually get stuck at a vertex, because D has no cycles. But the only vertex where it can get stuck is r, since every vertex $\neq r$ has an outgoing arc. So it will reach r. □

4.14.7. The Matrix-Tree Theorem: proof

We shall now prove the MTT. Here is our battle plan:

1. First, we prove the MTT in the case when each vertex $v \in V \setminus \{r\}$ has outdegree 1. In this case, after removing all the arcs with source r from D, we will see that D is either itself an arborescence or has a cycle; both cases will be easy to handle.
2. Then, we will prove the MTT in the slightly more general case when each $v \in V \setminus \{r\}$ has outdegree ≤ 1.
3. Finally, we will prove the MTT in the general case. This is done by strong induction on the # of arcs of D. Every time you have a vertex $v \in V \setminus \{r\}$ with outdegree > 1, you can pick such a vertex and color the outgoing arcs from it in red and blue in such a way that each color is used at least once. Then, you can consider the subdigraph of D obtained by removing

all the blue arcs (we call it D^{red}) and the subdigraph of D obtained by removing all the red arcs (we call it D^{blue}). Now apply the IH to D^{red} and to D^{blue}, and add together.

So let us begin with Step 1. We start with a very special case:

Lemma 4.14.10. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume

- that D has no cycles;
- that D has no arcs with source r;
- that each vertex $v \in V \setminus \{r\}$ has outdegree 1.

Then:

(a) The digraph D has a unique spanning arborescence rooted to r.

(b) If $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$, then the Laplacian L satisfies $\det(L_{\sim r, \sim r}) = 1$.

Proof. **(a)** The previous lemma shows that D itself is an arborescence rooted to r. So D itself is a spanning arborescence of D rooted to r. Any other spanning arborescence would have fewer arcs than D, which is impossible because of the $|A| = |V| - 1$ property of an arborescence. So the # of spanning arborescences rooted to r is 1.

(b) Let $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. WLOG assume that $r = n$ (otherwise, swap the labels r and n on D, and correspondingly permute the rows and the columns of L).

Moreover, WLOG assume that the remaining vertices of D are ordered by decreasing distance from $r = n$, that is,

$$d(1, r) \geq d(2, r) \geq \cdots \geq d(n-1, r) \geq d(n, r) = 0 \qquad (\text{since } n = r).$$

Note that paths are unique on D since each vertex has outdegree ≤ 1. Thus, if i and j are two vertices with $i < j$, then the digraph D has no arc from j to i (since the existence of such an arc would force $d(j, r) = d(i, r) + 1 > d(i, r)$, but $i < j$ leads to $d(i, r) \geq d(j, r)$). Therefore, in the definition of the Laplacian, we have $a_{j,i} = 0$ for all $i < j$, and therefore $L_{j,i} = 0$ for all $i < j$. That is, the matrix L is upper-triangular. Thus, its submatrix $L_{\sim r, \sim r}$ is also upper-triangular. Moreover, its diagonal entries are 1 (since all outdegrees are 1, and furthermore D has no cycles and thus no loops). So its determinant is $1 \cdot 1 \cdot \cdots \cdot 1 = 1$. In other words, $\det(L_{\sim r, \sim r}) = 1$. ☐

Next, we drop the "no cycles" condition:

Lemma 4.14.11. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that each vertex $v \in V \setminus \{r\}$ has outdegree 1. Then, the MTT holds for these D and r.

Proof. We assume WLOG that D has no arcs from r (because such arcs neither can appear in a spanning arborescence rooted to r, nor have any effect on $L_{\sim r,\sim r}$).

Again, we WLOG assume that $r = n$.

If D has no cycles, then the truth of the MTT follows from the previous lemma.

So we can WLOG assume that D has a cycle. Let $v_1, v_2, \ldots, v_m$ be the vertices of this cycle, in order, where $v_m = v_1$. Note that $r \notin \{v_1, v_2, \ldots, v_m\}$, since D has no arcs from r. Thus, D has no path from v_1 to r (since the outdegree condition does not allow going from v_1 to any vertex that is not in $\{v_1, v_2, \ldots, v_m\}$). Hence, D has no spanning arborescences rooted to r. Correspondingly, to prove that the MTT holds in this case, we must show that $\det(L_{\sim r,\sim r}) = 0$. But this can be done by showing that the v_1-th, v_2-th, ..., v_{m-1}-th rows of $L_{\sim r,\sim r}$ add up to the zero vector (which ensures that $L_{\sim r,\sim r}$ is singular). □

Now let us generalize a bit:

Lemma 4.14.12. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that each vertex $v \in V \setminus \{r\}$ has outdegree ≤ 1. Then, the MTT holds for these D and r.

Proof. If each vertex $v \in V \setminus \{r\}$ has outdegree 1, then this follows from the previous lemma. If some vertex $v \in V \setminus \{r\}$ has outdegree 0 (this is the only other option), then the MTT boils down to $0 = 0$ (indeed, $\det(L_{\sim r,\sim r}) = 0$ because the v-th row of L is 0). □

We are now ready for the general case, where the outdegrees are no longer required to be ≤ 1. First, we introduce a notation:

> Let M and N be two $n \times n$-matrices that agree in all but one row. That is, there exists some $j \in \{1, 2, \ldots, n\}$ such that for each $i \neq j$, we have
> $$(\text{the } i\text{-th row of } M) = (\text{the } i\text{-th row of } N).$$
> Then, we write $M \overset{j}{\equiv} N$, and we let $M \overset{j}{+} N$ be the $n \times n$-matrix that is obtained from M by adding the j-th row of N to the j-th row of M (while leaving all the other rows unchanged).

For example, if $M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$ and $N = \begin{pmatrix} a & b & c \\ d' & e' & f' \\ g & h & i \end{pmatrix}$, then $M \overset{2}{\equiv} N$

and

$$M \overset{2}{+} N = \begin{pmatrix} a & b & c \\ d+d' & e+e' & f+f' \\ g & h & i \end{pmatrix}.$$

A well-known property of determinants (the **multilinearity of the determinant**) says that if M and N are two $n \times n$-matrices and $j \in \{1, 2, \ldots, n\}$ is such that $M \overset{j}{\equiv} N$, then

$$\det \left(M \overset{j}{+} N \right) = \det M + \det N.$$

Proof of the MTT.. We proceed by strong induction on the # of arcs of D.

Induction step: Let $m \in \mathbb{N}$. Assume (as IH) that the MTT holds for all digraphs D with $< m$ arcs. We must now prove it for our digraph D with m arcs.

WLOG assume that $r = n$.

If each vertex $v \in V \setminus \{r\}$ has outdegree ≤ 1, then the MTT has already been proved above in our last lemma.

So assume WLOG that some vertex $v \in V \setminus \{r\}$ has outdegree > 1. Pick such a vertex v. Color the outgoing arcs from v in red and blue, making sure that each color is used at least once. All arcs with source $\neq v$ remain uncolored.

Now, let D^{red} be the subdigraph obtained from D by removing all blue arcs. This digraph D^{red} has fewer arcs than D; thus, its Laplacian L^{red} satisfies

$$\left(\text{\# of spanning arborescences of } D^{\text{red}} \text{ rooted to } r \right) = \det \left(L^{\text{red}}_{\sim r, \sim r} \right)$$

by the IH.

Now, let D^{blue} be the subdigraph obtained from D by removing all red arcs. This digraph D^{blue} has fewer arcs than D; thus, its Laplacian L^{blue} satisfies

$$\left(\text{\# of spanning arborescences of } D^{\text{blue}} \text{ rooted to } r \right) = \det \left(L^{\text{blue}}_{\sim r, \sim r} \right)$$

by the IH.

Now, I claim that

$$\begin{aligned} & (\text{\# of spanning arborescences of } D \text{ rooted to } r) \\ & = \left(\text{\# of spanning arborescences of } D^{\text{red}} \text{ rooted to } r \right) \\ & \qquad + \left(\text{\# of spanning arborescences of } D^{\text{blue}} \text{ rooted to } r \right), \end{aligned}$$

since each spanning arborescence of D rooted to r has $\deg^+ v = 1$ and thus uses either a red arc or a blue arc but never both at the same time.

On the other hand,

$$L^{\text{red}}_{\sim r, \sim r} \overset{v}{\equiv} L^{\text{blue}}_{\sim r, \sim r} \qquad \text{and} \qquad L^{\text{red}}_{\sim r, \sim r} \overset{v}{+} L^{\text{blue}}_{\sim r, \sim r} = L_{\sim r, \sim r}.$$

Thus, by the multilinearity of the determinant,

$$\det\left(L_{\sim r,\sim r}\right) = \det\left(L^{\text{red}}_{\sim r,\sim r}\right) + \det\left(L^{\text{blue}}_{\sim r,\sim r}\right).$$

Combine these identities and conclude that

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det\left(L_{\sim r,\sim r}\right).$$

□

See the references in the notes for various other proofs.

4.14.8. Application: Counting Eulerian circuits of K_n^{bidir}

Here is an easy consequence of the MTT:

Proposition 4.14.13. Let n be a positive integer. Pick any arc a of the multidigraph K_n^{bidir}. Then, the # of Eulerian circuits of K_n^{bidir} whose first arc is a is $n^{n-2} \cdot (n-2)!^n$.

Proof. WLOG the source of a is 1. The digraph K_n^{bidir} is balanced, and each of its vertices has outdegree $n-1$. So, by the BEST′ theorem,

$$\begin{aligned}
&\left(\text{\# of Eulerian circuits of } K_n^{\text{bidir}} \text{ whose first arc is } a\right) \\
&= \underbrace{\left(\text{\# of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right)}_{\substack{=n^{n-2} \\ \text{(as we saw above, since the spanning} \\ \text{arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1 \\ \text{are in bijection with the spanning trees of } K_n, \\ \text{and we saw that the latter are counted by } n^{n-2})}} \cdot \prod_{u=1}^{n} \left(\underbrace{\deg^+ u}_{=n-1} - 1\right)! \\
&= n^{n-2} \cdot \prod_{u=1}^{n} \underbrace{(n-1-1)!}_{=(n-2)!} = n^{n-2} \cdot (n-2)!^n.
\end{aligned}$$

□

In contrast, there is no good formula known for the # of Eulerian circuits of the undirected graph K_n. (See the OEIS A135388.)

4.15. The undirected Matrix-Tree Theorem

4.15.1. The theorem

The MTT (= Matrix-Tree Theorem) becomes simpler if we apply it to a digraph of the form G^{bidir}:

Theorem 4.15.1 (undirected MTT)**.** Let $G = (V, E, \varphi)$ be a multigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of the digraph G^{bidir}. Explicitly, this is the $n \times n$-matrix $L \in \mathbb{Z}^{n \times n}$ whose entries are given by

$$L_{i,j} = (\deg i) \cdot [i = j] - a_{i,j},$$

where $a_{i,j}$ is the # of edges that have endpoints i and j (with loops counted twice). Then:

(a) For any vertex r of G, we have

$$(\text{\# of spanning trees of } G) = \det (L_{\sim r, \sim r}) .$$

(b) Let t be an indeterminate. Expand the determinant $\det (tI_n + L)$ (where I_n is the $n \times n$ identity matrix) as a polynomial in t:

$$\det (tI_n + L) = c_n t^n + c_{n-1} t^{n-1} + \cdots + c_1 t^1 + c_0 t^0,$$

where $c_n, c_{n-1}, \ldots, c_0$ are numbers. (Note that this is just the characteristic polynomial of L up to substituting $-t$ for t and perhaps multiplying by $(-1)^n$. Some of its coefficients are $c_n = 1$ and $c_0 = \det L$ and $c_{n-1} = \operatorname{Tr} L$.) Then,

$$(\text{\# of spanning trees of } G) = \frac{1}{n} c_1.$$

(c) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$ (we can do this, since L is singular). Then,

$$(\text{\# of spanning trees of } G) = \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1}.$$

Proof. **(a)** Let r be a vertex of G. Then, we know that there is a bijection

$$\left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} \\ \to \{\text{spanning trees of } G\} .$$

Hence, by the bijection principle,

$$\begin{aligned} & (\text{\# of spanning trees of } G) \\ & = \left(\text{\# of spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right) \\ & = \det (L_{\sim r, \sim r}) \qquad (\text{by the directed MTT}) . \end{aligned}$$

This proves **(a)**.

(b) We claim that

$$c_1 = \sum_{r=1}^{n} \det\left(L_{\sim r, \sim r}\right).$$

Note that this is a purely linear-algebraic result, and holds for any $n \times n$-matrix instead of L. Once this claim is proved, part **(b)** will easily follow, since the claim entails

$$\begin{aligned}
\frac{1}{n} c_1 &= \frac{1}{n} \sum_{r=1}^{n} \underbrace{\det\left(L_{\sim r, \sim r}\right)}_{\substack{=(\#\text{ of spanning trees of } G) \\ (\text{by part } \textbf{(a)})}} \\
&= \frac{1}{n} \sum_{r=1}^{n} (\#\text{ of spanning trees of } G) \\
&= \frac{1}{n} \cdot n\, (\#\text{ of spanning trees of } G) = (\#\text{ of spanning trees of } G).
\end{aligned}$$

So we need to prove the claim

$$c_1 = \sum_{r=1}^{n} \det\left(L_{\sim r, \sim r}\right).$$

I give references to a rigorous proof in the notes, but here is just a proof-by-example: Let $n = 3$ and $L = \begin{pmatrix} a & b & c \\ a' & b' & c' \\ a'' & b'' & c'' \end{pmatrix}$. Then,

$$\begin{aligned}
\det\left(tI_n + L\right) &= \det \begin{pmatrix} t+a & b & c \\ a' & t+b' & c' \\ a'' & b'' & t+c'' \end{pmatrix} \\
&= t^3 + t^2 \left(a + b' + c''\right) \\
&\qquad + t \left(\det \begin{pmatrix} b' & c' \\ b'' & c'' \end{pmatrix} + \det \begin{pmatrix} a & c \\ a'' & c'' \end{pmatrix} + \det \begin{pmatrix} a & b \\ a' & b' \end{pmatrix}\right) \\
&\qquad + \det \begin{pmatrix} a & b & c \\ a' & b' & c' \\ a'' & b'' & c'' \end{pmatrix}.
\end{aligned}$$

So the t-coefficient (which we call c_1) is

$$\begin{aligned}
&\det \begin{pmatrix} b' & c' \\ b'' & c'' \end{pmatrix} + \det \begin{pmatrix} a & c \\ a'' & c'' \end{pmatrix} + \det \begin{pmatrix} a & b \\ a' & b' \end{pmatrix} \\
&= \sum_{r=1}^{n} \det\left(L_{\sim r, \sim r}\right).
\end{aligned}$$

The same logic works for any n.

(c) We need to show that $c_1 = \lambda_1 \lambda_2 \cdots \lambda_{n-1}$, where c_1 is as in part **(b)**. This is true for any matrix L with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $\lambda_n = 0$.

We have

$$\det(tI_n - L) = (t - \lambda_1)(t - \lambda_2) \cdots (t - \lambda_n),$$

so (by substituting $-t$ for t and scaling by $(-1)^n$) we get

$$\begin{aligned} \det(tI_n + L) &= (t + \lambda_1)(t + \lambda_2) \cdots (t + \lambda_n) \\ &= (t + \lambda_1)(t + \lambda_2) \cdots (t + \lambda_{n-1})\, t \qquad (\text{since } \lambda_n = 0). \end{aligned}$$

Comparing coefficients in front of t^1, we thus obtain

$$c_1 = \lambda_1 \lambda_2 \cdots \lambda_{n-1},$$

qed. □

4.15.2. Application: counting spanning trees of $K_{n,m}$

Laplacians of digraphs often have computable eigenvalues or computable minors, so the theorem above is often useful. A striking example is the hypercube graph Q_n, which is on the homework.

Here is a simpler example, in which we only need part **(a)** of the undirected MTT:

> **Exercise 2.** Let n and m be two positive integers. Let $K_{n,m}$ be the simple graph with $n + m$ vertices
>
> $$1, 2, \ldots, n, \qquad -1, -2, \ldots, -m,$$
>
> where two vertices i and j are adjacent if they have opposite signs (i.e., each positive vertex is adjacent to each negative vertex, but no two vertices of the same sign are adjacent).
>
> How many spanning trees does $K_{n,m}$ have?

Solution. If we rename the negative vertices $-1, -2, \ldots, -m$ as $n+1, n+2, \ldots, n+m$, then the Laplacian L of the digraph $K_{n,m}^{\text{bidir}}$ can be written in block-matrix notation as

$$L = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where

- A is a diagonal $n \times n$-matrix whose all diagonal entries are m;
- B is an $n \times m$-matrix whose all entries are -1;
- C is an $m \times n$-matrix whose all entries are -1;

- D is a diagonal $m \times m$-matrix whose all diagonal entries are n.

By part **(a)** of the undirected MTT, we have

$$(\text{\# of spanning trees of } K_{n,m}) = \det\left(L_{\sim r,\sim r}\right) \qquad \text{for any vertex } r.$$

We let $r = 1$. Then, the submatrix $L_{\sim r,\sim r} = L_{\sim 1,\sim 1}$ has the block-matrix form

$$L_{\sim r,\sim r} = \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix},$$

where

- $\widetilde{A}$ is a diagonal $(n-1) \times (n-1)$-matrix whose all diagonal entries are m;
- $\widetilde{B}$ is an $(n-1) \times m$-matrix whose all entries are -1;
- $\widetilde{C}$ is an $m \times (n-1)$-matrix whose all entries are -1;
- D is a diagonal $m \times m$-matrix whose all diagonal entries are n.

Fortunately, determinants of block matrices are often computable, at least when some of the blocks are invertible. Our life here is particularly easy since $\widetilde{A}$ and D are multiples of identity matrices: $\widetilde{A} = mI_{n-1}$ and $D = nI_m$. We perform a "blockwise row transformation" on the matrix $L_{\sim r,\sim r} = \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix}$ that subtracts the $\widetilde{C}\widetilde{A}^{-1}$-multiple of the first block row $\begin{pmatrix} \widetilde{A} & \widetilde{B} \end{pmatrix}$ from the second block row $\begin{pmatrix} \widetilde{C} & D \end{pmatrix}$. We get is

$$\begin{aligned} \det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix} &= \det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} - \widetilde{C}\widetilde{A}^{-1}\widetilde{A} & D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \end{pmatrix} \\ &= \det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ 0 & D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \end{pmatrix} \\ &= \det \widetilde{A} \cdot \det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right) \end{aligned}$$

(since the determinant of a block-triangular matrix can be computed by multiplying the determinants of its diagonal blocks).

Of course, $\det \widetilde{A} = m^{n-1}$, since $\widetilde{A}$ is a diagonal matrix. It remains to compute $\det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right)$. We know that $\widetilde{A}^{-1}$ is a diagonal matrix with diagonal entries m^{-1}; that is, $\widetilde{A}^{-1} = m^{-1}I_{n-1}$. The matrices $\widetilde{C}$ and $\widetilde{B}$ have all their entries equal to -1. So

$$\widetilde{C}\widetilde{A}^{-1}\widetilde{B} = m^{-1}\widetilde{C}\widetilde{B}$$

is the $m \times m$-matrix whose all entries equal $m^{-1}(n-1)$. Thus,

$$D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} = \begin{pmatrix} n - m^{-1}(n-1) & -m^{-1}(n-1) & -m^{-1}(n-1) & \cdots & -m^{-1}(n-1) \\ -m^{-1}(n-1) & n - m^{-1}(n-1) & -m^{-1}(n-1) & \cdots & -m^{-1}(n-1) \\ -m^{-1}(n-1) & -m^{-1}(n-1) & n - m^{-1}(n-1) & \cdots & -m^{-1}(n-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -m^{-1}(n-1) & -m^{-1}(n-1) & -m^{-1}(n-1) & \cdots & n \end{pmatrix}$$

is the $m \times m$-matrix whose all diagonal entries are $n - m^{-1}(n-1)$ and whose all off-diagonal entries are $-m^{-1}(n-1)$. What is its determinant? We have already computed the determinant of another matrix of this form; let us now deal with the general case:

Proposition 4.15.2. Let $n \in \mathbb{N}$. Let x and a be two numbers. Then,

$$\det \underbrace{\begin{pmatrix} x & a & a & \cdots & a \\ a & x & a & \cdots & a \\ a & a & x & \cdots & a \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a & a & a & \cdots & x \end{pmatrix}}_{\substack{\text{the } n\times n\text{-matrix} \\ \text{whose diagonal entries are } x \\ \text{and whose off-diagonal entries are } a}} = (x + (n-1)a)(x-a)^{n-1}.$$

This proposition can be proved using similar reasoning as the determinant in our proof of Cayley's formula; see the notes for some more general facts.

Anyway, applying this proposition to m, $n - m^{-1}(n-1)$ and $-m^{-1}(n-1)$ instead of n, x and a, we obtain

$$\begin{aligned} \det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right) &= \underbrace{\left(\left(n - m^{-1}(n-1)\right) + (m-1)\left(-m^{-1}(n-1)\right)\right)}_{=1} \\ &\qquad \cdot \left(\underbrace{\left(n - m^{-1}(n-1)\right) - \left(-m^{-1}(n-1)\right)}_{=n}\right)^{m-1} \\ &= n^{m-1}. \end{aligned}$$

So

$$\det\begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix} = \underbrace{\det \widetilde{A}}_{=m^{n-1}} \cdot \underbrace{\det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right)}_{=n^{m-1}} = m^{n-1}n^{m-1}.$$

$\square$

Thus we have proved:

Theorem 4.15.3. Let n and m be two positive integers. Then,

$$(\text{\# of spanning trees of } K_{n,m}) = m^{n-1}n^{m-1}.$$

4.16. de Bruijn sequences

4.16.1. Definition

A little puzzle: What is special about the periodic sequence

$$|| : \ 0000\ 1111\ 0110\ 0101 \ : || \qquad ?$$

One nice property is that if you slide a "length-4 window" along this sequence, then you will get all possible bitstrings of length 4 depending on the position of the window, and these bitstrings never repeat until you move 16 steps to the right. Just see:

0000,	0001,	0011,	0111,	1111,
1110,	1101,	1011,	0110,	1100,
1001,	0010,	0101,	1010,	0100,
1000.				

Can we find such nice sequences for any window length, not just 4 ?
For instance, for window length 3, we have

$$|| : \ 000\ 111\ 01 \ : || \ .$$

What about higher window length? Is there always a periodic sequence of period 2^n that gives all possible length-n bitstrings by sliding a window along it?
Also, does this work if bits are replaced by elements of a larger set?
Let us give this a name:

Definition 4.16.1. Let n and k be two positive integers, and let K be a k-element set.

A **de Bruijn sequence** of order n on K means a k^n-tuple $(c_0, c_1, \ldots, c_{k^n-1})$ of elements of K such that

(A) for each n-tuple $(a_1, a_2, \ldots, a_n) \in K^n$ of elements of K, there is a **unique** $r \in \{0, 1, \ldots, k^n - 1\}$ such that

$$(a_1, a_2, \ldots, a_n) = (c_r, c_{r+1}, \ldots, c_{r+n-1}).$$

Here, the indices under the letter "c" are understood to be periodic modulo k^n; that is, we set $c_{q+k^n} = c_q$ for each $q \in \mathbb{Z}$.

For example, for $n = 2$ and $k = 3$ and $K = \{0, 1, 2\}$, the 9-tuple

$$(0, 0, 1, 1, 2, 2, 0, 2, 1)$$

is a de Bruijn sequence of order n on K, because if we label its entries as $c_0, c_1, \ldots, c_8$ (and extend the indices periodically), then

$$\begin{aligned} (0,0) &= (c_0, c_1), & (0,1) &= (c_1, c_2), & (0,2) &= (c_6, c_7), \\ (1,0) &= (c_8, c_9), & (1,1) &= (c_2, c_3), & (1,2) &= (c_3, c_4), \\ (2,0) &= (c_5, c_6), & (2,1) &= (c_7, c_8), & (2,2) &= (c_4, c_5). \end{aligned}$$

In our previous notation, this is the periodic sequence $||:$ 00 11 22 02 1 $:||$.

4.16.2. Existence of de Bruijn sequences

It turns out that de Bruijn sequences always exist:

> **Theorem 4.16.2** (de Bruijn, Sainte-Marie)**.** Let n and k be positive integers. Let K be a k-element set. Then, a de Bruijn sequence of order n on K exists.

Proof. It looks reasonable to apporach this using a digraph. For example, we can define a digraph whose vertices are the n-tuples in K^n, and that has an arc from one n-tuple i to another n-tuple j if j can be obtained from i by dropping the first entry and adding a new entry at the end. Then, a de Bruijn sequence (of order n on K) is the same as a Hamiltonian cycle of this digraph.

Sadly, this does not help – we do not have any criteria that would guarantee the existence of a Hamiltonian cycle here.

However, let us do something counterintuitive: We reinterpret the problem in terms of Eulerian circuits instead of Hamiltonian cycles.

For this, we need a different digraph. Namely, we define D to be the multidigraph $\left(K^{n-1}, K^n, \psi\right)$, where the map $\psi : K^n \to K^{n-1} \times K^{n-1}$ is given by the formula

$$\psi (a_1, a_2, \ldots, a_n) = ((a_1, a_2, \ldots, a_{n-1}),\ (a_2, a_3, \ldots, a_n)).$$

Thus, the vertices of D are the $(n-1)$-tuples (not the n-tuples!) of elements of K, whereas the arcs are the n-tuples, and each such arc $(a_1, a_2, \ldots, a_n)$ has source $(a_1, a_2, \ldots, a_{n-1})$ and target $(a_2, a_3, \ldots, a_n)$. Hence, there is an arc from each $(n-1)$-tuple i to each $(n-1)$-tuple j that is obtained from i by dropping the first entry and inserting a new entry at the end. (Careful with the $n = 1$ case, but that case is trivial anyway.)

Some things to notice about D:

1. The multidigraph D is strongly connected.

[*Proof:* To get from a vertex $i = (i_1, i_2, \ldots, i_{n-1})$ to a vertex $j = (j_1, j_2, \ldots, j_{n-1})$, we can simply slide all the entries of j into i, pushing out the existing entries of i:

$$\begin{aligned} i = (i_1, i_2, \ldots, i_{n-1}) &\to (i_2, i_3, \ldots, i_{n-1}, j_1) \\ &\to (i_3, i_4, \ldots, i_{n-1}, j_1, j_2) \to \cdots \\ &\to (j_1, j_2, \ldots, j_{n-1}) = j. \end{aligned}$$

Note that this is a walk of length $n-1$, and is the unique walk from i to j that have length $n-1$. Thus, the # of walks from i to j that have length $n-1$ is 1.]

2. Thus, D is weakly connected.
3. The multidigraph D is balanced, and in fact each vertex of D has outdegree k and indegree k.
4. The digraph D has an Eulerian circuit.

   [This follows either from the directed Euler–Hierholzer theorem or from the BEST theorem.]

So we know that D has an Eulerian circuit $\mathbf{c}$. This leads to a de Bruijn sequence as follows:

Let $p_0, p_1, \ldots, p_{k^n-1}$ be the arcs of $\mathbf{c}$. Extend the subscripts periodically mod k^n. Let x_i be the first entry of p_i for each $i \in \mathbb{Z}$. Then, I claim that $(x_0, x_1, \ldots, x_{k^n-1})$ is a de Bruijn sequence. Indeed, the "sliding windows" $(x_i, x_{i+1}, \ldots, x_{i+n-1})$ of this sequence are precisely the n-tuples p_i, because

- x_i is the first entry of p_i;
- x_{i+1} is the first entry of p_{i+1}, but the "shift left and insert" rule for the arcs means that this is the second entry of p_i;
- x_{i+2} is the first entry of p_{i+2}, but that's the second entry of p_{i+1} and therefore the third entry of p_i;
- and so on.

Since the n-tuples $p_0, p_1, \ldots, p_{k^n-1}$ contain each n-tuple in K^n exactly once (that's what being an Eulerian circuit means), we thus conclude that the "sliding windows" $(x_i, x_{i+1}, \ldots, x_{i+n-1})$ also contain each n-tuple in K^n exactly once. Thus, $(x_0, x_1, \ldots, x_{k^n-1})$ is a de Bruijn sequence. □

There is much more to say about de Bruijn sequences. There are also several variations on de Bruijn sequences.

4.16.3. Counting de Bruijn sequences

Now that we have proved that de Bruijn sequences exist, maybe we can also count them?

It makes sense to apply the MTT theorem to the digraph D that we used above. Note that D is balanced but not of the form G^{bidir}. Nevertheless, something very close to the undirected MTT holds:

Theorem 4.16.3 (balanced MTT)**.** Let $D = (V, A, \psi)$ be a balanced multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of the digraph D. Then:

(a) For any vertex r of D, we have

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det\left(L_{\sim r, \sim r}\right).$$

Moreover, this number does not depend on r.

(b) Let t be an indeterminate. Expand the determinant $\det(tI_n + L)$ (where I_n is the $n \times n$ identity matrix) as a polynomial in t:

$$\det(tI_n + L) = c_n t^n + c_{n-1} t^{n-1} + \cdots + c_1 t^1 + c_0 t^0,$$

where $c_n, c_{n-1}, \ldots, c_0$ are numbers. (Note that this is just the characteristic polynomial of L up to substituting $-t$ for t and perhaps multiplying by $(-1)^n$. Some of its coefficients are $c_n = 1$ and $c_0 = \det L$ and $c_{n-1} = \operatorname{Tr} L$.) Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \frac{1}{n} c_1.$$

(c) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$ (we can do this, since L is singular). Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1}.$$

(d) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$ (we can do this, since L is singular). If all vertices of D have outdegree > 0, then

$$(\text{\# of Eulerian circuits of } D) = |A| \cdot \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1} \cdot \prod_{u \in V} \left(\deg^+ u - 1\right)!.$$

(If you identify an Eulerian circuit with its cyclic rotations, then the $|A|$ factor falls away.)

Proof. **(a)** This is the MTT. The independence has been done before.

(b), (c) as in the undirected graph case.

(d) Follows from **(c)** using the BEST theorem. (Note that we are not fixing the first arc of our Eulerian circuits, so we need to sum the results over all $a \in A$, which leads to the $|A|$ factor.) $\square$

Now, let us apply this to counting de Bruijn sequences. In our proof of existence of de Bruijn sequences, we constructed a map

$$\begin{aligned}\{\text{Eulerian circuits of } D\} &\to \{\text{de Bruijn sequences of order } n \text{ on } K\}, \\ \mathbf{c} &\mapsto (\text{the sequence of first entries of the arcs of } \mathbf{c}).\end{aligned}$$

It turns out that this map is a bijection (why?). Thus,

$$\begin{aligned}&(\text{\# of de Bruijn sequences of order } n \text{ on } K) \\ &= (\text{\# of Eulerian circuits of } D) \\ &= |K^n| \cdot \frac{1}{k^{n-1}} \cdot \lambda_1\lambda_2\cdots\lambda_{k^{n-1}-1} \cdot \prod_{u \in K^{n-1}} \Big(\underbrace{\deg^+ u}_{=k} - 1\Big)! \\ &\qquad \left(\begin{array}{c}\text{by part \textbf{(d)} of the balanced MTT,} \\ \text{noting that } D \text{ has } k^{n-1} \text{ vertices}\end{array}\right) \\ &= \underbrace{k^n \cdot \frac{1}{k^{n-1}}}_{=k} \cdot \lambda_1\lambda_2\cdots\lambda_{k^{n-1}-1} \cdot \underbrace{\prod_{u \in K^{n-1}} (k-1)!}_{=(k-1)!^{k^{n-1}}} \\ &= k \cdot \lambda_1\lambda_2\cdots\lambda_{k^{n-1}-1} \cdot (k-1)!^{k^{n-1}}.\end{aligned}$$

To conclude this computation, we need to find the eigenvalues of L. How can we do this?

The Laplacian L of our digraph D is a $k^{n-1} \times k^{n-1}$-matrix whose rows and columns are indexed by the $(n-1)$-tuples in K^{n-1}. Strictly speaking, we should reindex them by $1, 2, \ldots, k^{n-1}$, but let us not do this here.

Let C be the adjacency matrix of the digraph D; this is a $k^{n-1} \times k^{n-1}$-matrix whose (i, j)-th entry is the # of arcs with source i and target j. In particular,

$$\operatorname{Tr} C = (\text{\# of loops in } D) = k.$$

However, C is closely related to L: Namely,

$$L = \Delta - C, \qquad \text{where } \Delta = \operatorname{diag}\left(\deg^+ 1,\ \deg^+ 2,\ \ldots,\ \deg^+\left(k^{n-1}\right)\right).$$

Since all outdegrees in D are just k, we have $\Delta = k \cdot I_{k^{n-1}}$, so that

$$L = k \cdot I_{k^{n-1}} - C.$$

Thus, finding the eigenvalues of L is tantamount to finding the eigenvalues of C. Now let us do this.

Experiments may suggest that the eigenvalues of C are $\underbrace{0,0,\ldots,0}_{k^{n-1}-1 \text{ many zeroes}},k$.

But C does not have rank 1.

To prove this, we do something underhanded: We observe that

$$C^{n-1} = J,$$

where J is the $k^{n-1} \times k^{n-1}$-matrix whose all entries are 1. Indeed, by what we know about adjacency matrices, each entry of C^{n-1} is given by

$$\begin{aligned}\left(C^{n-1}\right)_{i,j} &= (\text{\# of walks from } i \text{ to } j \text{ having length } n-1 \text{ in } D) \\ &= 1\end{aligned}$$

because we saw last time that there is only one such walk.

So $C^{n-1} = J = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{k^{n-1}\times k^{n-1}}$, which has rank 1 and thus has $k^{n-1}-1$ eigenvalues equal to 0. What does this mean for C ?

The spectral mapping theorem says that if a matrix M has eigenvalues $\mu_1, \mu_2, \ldots, \mu_m$, then its r-th power M^r has eigenvalues $\mu_1^r, \mu_2^r, \ldots, \mu_m^r$. Thus, the eigenvalues of C^{n-1} are the $(n-1)$-st powers of the eigenvalues of C. Since all but one of the former eigenvalues are 0, this entails that all but one of the latter eigenvalues are 0. In other words, the eigenvalues of C are $0,0,\ldots,0,?$ for some unknown eigenvalue ?.

Thus, the eigenvalues of $L = k \cdot I_{k^{n-1}} - C$ are

$$\underbrace{k-0,\ k-0,\ \ldots,\ k-0}_{\text{just } k,k,\ldots,k},\quad \underbrace{k-\ ?}_{\substack{\text{this must be 0} \\ \text{since } L \text{ is singular}}},$$

that is,

$$\underbrace{k,k,\ldots,k}_{k^{n-1}-1 \text{ times}},0.$$

Altogether, we now get

$$\begin{aligned}&(\text{\# of de Bruijn sequences of order } n \text{ on } K) \\ &= k \cdot \underbrace{\lambda_1\lambda_2\cdots\lambda_{k^{n-1}-1}}_{=kk\cdots k=k^{k^{n-1}-1}} \cdot (k-1)!^{k^{n-1}} \\ &= \underbrace{k \cdot k^{k^{n-1}-1}}_{=k^{k^{n-1}}} \cdot (k-1)!^{k^{n-1}} = k^{k^{n-1}} \cdot (k-1)!^{k^{n-1}} \\ &= (k\cdot(k-1)!)^{k^{n-1}} = k!^{k^{n-1}}.\end{aligned}$$

So we have proved:

Theorem 4.16.4. Let n and k be two positive integers. Let K be a k-element set. Then,

$$(\text{\# of de Bruijn sequences of order } n \text{ on } K) = k!^{k^{n-1}}.$$

Recently this has been proved combinatorially.

4.17. More on Laplacians

See the notes for some references:

- electrical networks (Kirchhoff's work),
- spectral layout (drawing graphs).

4.18. On the left nullspace of the Laplacian

For a square matrix, the dimension of the left nullspace equals the dimension of the right nullspace (by the rank-nullity theorem, since row rank = column rank). So when you know that the right nullspace contains a nonzero vector, you can conclude that so does the left nullspace. But what is this vector?

This question can be asked in particular for the Laplacian L of a digraph D: We know that $Le = 0$ where $e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$; but can we also find a nonzero row vector f such that $fL = 0$? Yes, thanks to the following theorem:

Theorem 4.18.1 (harmonic vector theorem for Laplacians)**.** Let $D = (V, A, \psi)$ be a multidigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

For each $r \in V$, let $\tau(D, r)$ be the # of spanning arborescences of D rooted to r.

Let f be the row vector $(\tau(D,1),\ \tau(D,2),\ \ldots,\ \tau(D,n))$. Then, $fL = 0$.

Note that this f is nonzero when D has a to-root; otherwise nonzero vectors in the left nullspace of L can be constructed by applying this theorem to a sink component of D. (See more in the notes.)

The theorem is "essentially" an explicit formula for the steady state of an irreducible Markov chain with rational probabilities.

Also in the notes is a proof of the theorem. Along the way, the proof shows the following variant of the MTT:

Theorem 4.18.2 (Matrix-Tree Theorem, off-diagonal version)**.** Let $D = (V, A, \psi)$ be a multidigraph. Assume that $V = \{1, 2, \ldots, n\}$ for some positive integer n.

Let L be the Laplacian of D. Let r and s be any two vertices of D. Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = (-1)^{r+s} \det (L_{\sim r, \sim s}) .$$

Proof idea. The Laplacian L has the property that the sum of all its columns is 0. This can be shown to entail that its cofactors are constant along each row. So $(-1)^{r+s} \det (L_{\sim r, \sim s}) = \det (L_{\sim r, \sim r})$, and thus the off-diagonal version of the MTT follows from the original version. □

4.19. A weighted Matrix-Tree Theorem

4.19.1. Definitions

We have so far been **counting** arborescences. A natural generalization of counting is **weighted counting** – i.e., you assign a certain number (a "weight") to each arborescence (or whatever you want to count), and then you **sum** the weights of all the arborescences. This generalizes counting, which you obtain if all the weights equal 1.

Completely random weights do not usually yield interesting results. However, some choices of weights do. Let us see what we get if we assign a weight to each **arc** of our digraph, and then define the weight of an arborescences to be the **product** of the weights of the arcs in it.

Definition 4.19.1. Let $D = (V, A, \psi)$ be a multidigraph.

Let $\mathbb{K}$ be a commutative ring. Assume that an element $w_a \in \mathbb{K}$ is assigned to each arc $a \in A$. We call this w_a the **weight** of the arc a.

- **(a)** For any two vertices $i, j \in V$, we let $a^w_{i,j}$ be the sum of the weights of all arcs from i to j.
- **(b)** For any vertex $i \in V$, we define the **weighted outdegree** $\deg^{+w} i$ of i by

$$\deg^{+w} i := \sum_{\substack{a \in A; \\ a \text{ has source } i}} w_a .$$

- **(c)** If B is a subdigraph of D, then the **weight** $w(B)$ of B is defined to be the product

$$\prod_{a \text{ is an arc of } B} w_a$$

  of the weights of all arcs of B.

(d) Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. The **weighted Laplacian** of D (with respect to the weights w_a) is defined to be the $n \times n$-matrix $L^w \in \mathbb{K}^{n \times n}$ defined by

$$L^w_{i,j} = \left(\deg^{+w} i\right) \cdot [i = j] - a^w_{i,j} \qquad \text{for all } i, j \in V.$$

Theorem 4.19.2 (weighted MTT)**.** Let $D = (V, A, \psi)$ be a multidigraph.

Let $\mathbb{K}$ be a commutative ring. Assume that an element $w_a \in \mathbb{K}$ is assigned to each arc $a \in A$. We call this w_a the **weight** of the arc a.

Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. Let L^w be the weighted Laplacian of D.

Let r be a vertex of D. Then,

$$\sum_{\substack{B \text{ is a spanning} \\ \text{arborescence} \\ \text{of } D \text{ rooted to } r}} w(B) = \det\left(L^w_{\sim r, \sim r}\right).$$

This generalizes the original MTT, which we can recover by setting all $w_a := 1$.

But we can also go the other way round, deriving the weighted MTT from the original MTT. This is what we will do next.

4.19.2. The polynomial identity trick

First we recall a standard result in algebra, known as the **principle of permanence of polynomial identities** or the **polynomial identity trick**, in the following incarnation:

Theorem 4.19.3. Let $P(x_1, x_2, \ldots, x_m)$ and $Q(x_1, x_2, \ldots, x_m)$ be two polynomials with integer coefficients in several indeterminates $x_1, x_2, \ldots, x_m$. Assume that the equality

$$P(k_1, k_2, \ldots, k_m) = Q(k_1, k_2, \ldots, k_m)$$

holds for every m-tuple $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$ of nonnegative integers. Then, $P(x_1, x_2, \ldots, x_m)$ and $Q(x_1, x_2, \ldots, x_m)$ are equal as polynomials, so that the above equality holds not only for $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$ but also for all $(k_1, k_2, \ldots, k_m) \in \mathbb{R}^m$ or more generally for all $(k_1, k_2, \ldots, k_m) \in \mathbb{K}^m$ for any commutative ring $\mathbb{K}$.

This is often summarized as "in order to prove that two polynomials are equal, it suffices to show that they agree on all nonnegative integer points".

For instance, if you can prove the identity

$$(x+y)^4 + (x-y)^4 = 2x^4 + 12x^2y^2 + 2y^4$$

for all $x, y \in \mathbb{N}$, then it automatically follows for all $x, y \in \mathbb{R}$ and for all $x, y \in \mathbb{K}$ for any commutative ring $\mathbb{K}$.

To prove the theorem, start by showing the $m = 1$ case using the easy half of the FTA. Then, induct on m to cover the general case. To go from m to $m+1$, consider the new variable as the only variable while the others as constants.

4.19.3. Proof of the weighted MTT

We can now deduce the weighted MTT from the original MTT:

Proof of the weighted MTT.. The claim of the weighted MTT (for fixed D and r) is an equality between two polynomials in the arc weights w_a. Therefore, thanks to the polynomial identity trick, it suffices to prove this equality in the case when all arc weights w_a are nonnegative integers. So we WLOG assume that we are in this case.

Now, let us replace each arc a of D by w_a many copies of the arc a (having the same source and target as a). The result is a new digraph D'.

It is easy to see that

$$\deg^+_{D'} i = \deg^{+w}_D i \qquad \text{for each } i \in V.$$

Thus, the weighted Laplacian L^{+w} of D equals the usual Laplacian L of D'.

On the other hand, the sum $\sum\limits_{\substack{B \text{ is a spanning} \\ \text{arborescence} \\ \text{of } D \text{ rooted to } r}} w(B)$ is just the # of spanning arborescences of D' rooted to r. Thus, the weighted MTT for D is simply the usual MTT for D'. □

There are other proofs of the weighted MTT as well; in particular we can get by simply modifying the proof of the original MTT.

4.19.4. Application: Counting trees by their degrees

Exercise 3. Let $n \geq 2$ be an integer, and let $d_1, d_2, \ldots, d_n$ be n positive integers. An n**-tree** shall mean a simple graph with vertex set $\{1, 2, \ldots, n\}$ that is a tree. We know that there are n^{n-2} many n-trees. Now, how many of these n-trees have the property that

$$\deg i = d_i \qquad \text{for each } i \in \{1, 2, \ldots, n\} \ ?$$

Solution. The n-trees are just the spanning trees of K_n.

To incorporate the $\deg i = d_i$ condition into our count, we use a generating function. So let us **not** fix the $d_1, d_2, \ldots, d_n$, but rather consider the polynomial

$$P(x_1, x_2, \ldots, x_n) := \sum_{T \text{ is an } n\text{-tree}} x_1^{\deg 1} x_2^{\deg 2} \cdots x_n^{\deg n}$$

in n indeterminates $x_1, x_2, \ldots, x_n$ (where $\deg i := \deg_T i$). Then, the $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$-coefficient of this polynomial $P(x_1, x_2, \ldots, x_n)$ is the # of n-trees T satisfying the property that

$$\deg i = d_i \qquad \text{for each } i \in \{1, 2, \ldots, n\}.$$

Let us assign to each edge ij of K_n the weight $w_{ij} := x_i x_j$. Then, the definition of $P(x_1, x_2, \ldots, x_n)$ rewrites as

$$P(x_1, x_2, \ldots, x_n) = \sum_{T \text{ is a } n\text{-tree}} w(T),$$

where $w(T)$ is the product of the weights of all edges of T.

Let us assign the same weights to the arfcs of the digraph K_n^{bidir}. That is, the two arcs $(ij, 1)$ and $(ij, 2)$ corresponding to an edge ij of K_n shall both have the weight

$$w_{(ij,1)} = w_{(ij,2)} = w_{ij} = x_i x_j.$$

As we already have seen, we can replace spanning trees of K_n by spanning arborescences of K_n^{bidir} rooted to 1, since there is a bijection from the former to the latter. Thus,

$$\begin{aligned} &(\# \text{ of spanning trees of } K_n) \\ &= \left(\# \text{ of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right). \end{aligned}$$

Since this bijection preserves weights, it also gives

$$\sum_{\substack{T \text{ is a spanning} \\ \text{tree of } K_n, \\ \text{i.e., an } n\text{-tree}}} w(T) = \sum_{\substack{B \text{ is a spanning} \\ \text{arborescence of } K_n^{\text{bidir}} \\ \text{rooted to } 1}} w(B).$$

Now we can compute the RHS using the weighted MTT, since the weighted

Laplacian L^w of K_n^{bidir} is the $n \times n$-matrix with entries

$$\begin{aligned} L^w_{i,j} &= \left(\deg^{+w} i\right) \cdot [i = j] - a^w_{i,j} \\ &= \begin{cases} \deg^{+w} i - a^w_{i,j}, & \text{if } i = j; \\ -a^w_{i,j}, & \text{if } i \neq j \end{cases} \\ &= \begin{cases} \deg^{+w} i, & \text{if } i = j; \\ -a^w_{i,j}, & \text{if } i \neq j \end{cases} \qquad \text{(since } K_n \text{ has no loops)} \\ &= \begin{cases} x_i \left(x_1 + x_2 + \cdots + x_n\right) - x_i x_j, & \text{if } i = j; \\ -x_i x_j, & \text{if } i \neq j \end{cases} \\ &= [i = j] \cdot x_i \left(x_1 + x_2 + \cdots + x_n\right) - x_i x_j \\ &= x_i \cdot \left([i = j] \left(x_1 + x_2 + \cdots + x_n\right) - x_j\right). \end{aligned}$$

Now we can find the minor $\det\left(L^w_{\sim 1, \sim 1}\right)$ of this matrix pretty easily:

$$\det\left(L^w_{\sim 1, \sim 1}\right) = x_1 x_2 \cdots x_n \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}.$$

Summarizing what we have done so far, we get

$$\begin{aligned} P\left(x_1, x_2, \ldots, x_n\right) &= \sum_{T \text{ is a } n\text{-tree}} w\left(T\right) \\ &= \sum_{\substack{B \text{ is a spanning} \\ \text{arborescence of } K_n^{\text{bidir}} \\ \text{rooted to } 1}} w\left(B\right) \\ &= \det\left(L^w_{\sim 1, \sim 1}\right) \qquad \text{(by the weighted MTT)} \\ &= x_1 x_2 \cdots x_n \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}. \end{aligned}$$

Thus,

$$\begin{aligned} &\left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient in } P\left(x_1, x_2, \ldots, x_n\right)\right) \\ &= \left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient in } x_1 x_2 \cdots x_n \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}\right) \\ &= \left(\text{the } x_1^{d_1 - 1} x_2^{d_2 - 1} \cdots x_n^{d_n - 1}\text{-coefficient in } \left(x_1 + x_2 + \cdots + x_n\right)^{n-2}\right) \\ &= \binom{n-2}{d_1 - 1,\ d_2 - 1,\ \ldots,\ d_n - 1}, \end{aligned}$$

where the **multinomial coefficient** $\dbinom{a}{b_1, b_2, \ldots, b_n}$ is defined to be $\dfrac{a!}{b_1! b_2! \cdots b_n!}$ when $b_1 + b_2 + \cdots + b_n = a$ and is defined to be 0 otherwise (so $\dbinom{a}{b_1, b_2, \ldots, b_n}$ is the $x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ in $\left(x_1 + x_2 + \cdots + x_n\right)^a$). Thus we have proved: □

Theorem 4.19.4 (refined Cayley's formula)**.** Let $n \geq 2$ be an integer, and let $d_1, d_2, \ldots, d_n$ be n positive integers. Then, the # of n-trees with the property that

$$\deg i = d_i \qquad \text{for each } i \in \{1, 2, \ldots, n\}$$

is the multinomial coefficient

$$\binom{n-2}{d_1 - 1,\ d_2 - 1,\ \ldots,\ d_n - 1} = \begin{cases} \dfrac{(n-2)!}{(d_1-1)!\,(d_2-1)!\cdots(d_n-1)!}, & \text{if } d_1 + d_2 + \cdots + d_n = 2n - \\ 0, & \text{otherwise.} \end{cases}$$

4.19.5. The weighted harmonic vector theorem

The harmonic vector theorem also has a weighted version. Note that every $n \times n$-matrix whose column vectors sum to the zero vector is the weighted Laplacian of some digraph (actually of K_n^{bidir}). So the theorem gives an explicit left nullspace vector for any such matrix. Cramer's rule does the same, of course, but the harmonic vector theorem has predictable signs. This can be used to prove various facts, such as the following classical result from probability theory:

Theorem 4.19.5. Let n be a positive integer. Let $p = (p_{i,j})_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$ be an $n \times n$-matrix whose entries are real and nonnegative. Assume that

$$\sum_{j=1}^{n} p_{i,j} = 1 \qquad \text{for each } i \in \{1, 2, \ldots, n\}.$$

(Matrices P satisfying these conditions are called **stochastic**.)

Let D be the simple digraph (V, A) with vertex set $V = \{1, 2, \ldots, n\}$ and with arc set $A = \{(i,j) \mid p_{i,j} > 0\}$. Assume that this digraph D is strongly connected. (Matrices P satisfying this condition are called **irreducible**.) Then:

(a) There is a unique row vector $x \in \mathbb{R}^{n \times 1}$ such that $xP = x$ and such that the sum of all entries of x is 1. (This vector x is called the **stationary distribution** or the **steady state** of the Markov chain defined by P.)

(b) The entries of this vector x are positive reals.

(c) If all the $p_{i,j}$ are rational, then so are the entries of x.

Proof idea. Assign to each arc $a = (i,j)$ of D the weight $w_a = p_{i,j}$. Then, the weighted Laplacian L^w of this D will be $I_n - P$. Then, the row vector x we are looking for must satisfy $xL^w = 0$. The weighted harmonic vector theorem gives us such an x: namely, $x = (\tau^w(D,1),\ \tau^w(D,2),\ \ldots,\ \tau^w(D,n))$, where

$\tau^w(D,r)$ is the # of spanning arborescences of D rooted to r. The positivity of $\tau^w(D,r)$ is clear from the strong connectivity of D. The rationality is completely obvious. □

5. Colorings

Now to something different: Let's color the vertices of a graph!

This is a serious course, so our colors are positive integers.

5.1. Definition

Definition 5.1.1. Let $G = (V, E, \varphi)$ be a multigraph. Let $k \in \mathbb{N}$.

(a) A k-**coloring** of G means a map $f : V \to \{1, 2, \ldots, k\}$. Given such a k-coloring f, we refer to the numbers $1, 2, \ldots, k$ as the **colors**, and we refer to each value $f(v)$ as the **color** of the vertex v in f.

(b) A k-coloring f of G is said to be **proper** if no two adjacent vertices of G have the same color (i.e., no edge of G has endpoints u and v satisfying $f(u) = f(v)$).

Example 5.1.2. See notes / done on the whiteboard.

Example 5.1.3. Let $n \in \mathbb{N}$. The n-hypercube graph Q_n has a proper 2-coloring: Namely, the map

$$f : \{0,1\}^n \to \{1,2\},$$
$$(a_1, a_2, \ldots, a_n) \mapsto \begin{cases} 1, & \text{if } a_1 + a_2 + \cdots + a_n \text{ is even;} \\ 2, & \text{if } a_1 + a_2 + \cdots + a_n \text{ is odd} \end{cases}$$

is a proper 2-coloring of Q_n.

Example 5.1.4. See notes for the grid graph $P_n \times P_m$.

Finding proper 3-colorings, or even just determining whether they exist, is an NP-complete problem.

5.2. 2-colorings

In contrast, the existence of proper 2-colorings is a much simpler question. Here is a nice criterion:

Theorem 5.2.1 (2-coloring equivalence theorem)**.** Let G be a multigraph. Then, the following statements are equivalent:

- **Statement B1:** The graph G has a proper 2-coloring.
- **Statement B2:** The graph G has no cycles of odd length.
- **Statement B3:** The graph G has no circuits of odd length.

Proof. B3$\Longrightarrow$B2: Obvious (cycles are circuits).

B1$\Longrightarrow$B3: Walk along the circuit and watch the color of each vertex flip every step.

B3$\Longrightarrow$B1: WLOG G is connected. Then, pick a vertex r, and color each vertex v of G with the parity of $d(v,r)$ (that is, let v have color 1 if $d(v,r)$ is odd and color 2 if $d(v,r)$ is even). I claim that this is a proper 2-coloring. Indeed, if v and w are two adjacent vertices of the same color, then we can splice together the shortest paths from r to v and w and the edge joining v with w and obtain a circuit of odd length, which contradicts B3.

B2$\Longrightarrow$B3: This was a HW exercise essentially. Better: Any odd-length walk from u to v contains an odd-length path from u to v or an odd-length cycle (or both). This is proved by shrinking the walk through removing minimum-length cycles from it. □

A graph G that satisfies the three equivalent statements B1, B2 and B3 above is sometimes called a "bipartite graph". However, we will rather use this language for a graph **equipped with** a proper 2-coloring. A graph having a proper 2-coloring usually has more than one such. In fact we can compute the total number:

Proposition 5.2.2. Let G be a multigraph that has a proper 2-coloring. Then, G has exactly $2^{\operatorname{conn} G}$ proper 2-colorings.

Interestingly, there is a version of the 2-coloring equivalence theorem for directed graphs:

Theorem 5.2.3. Let D be a strongly connected multidigraph. Then, the following statements are equivalent:

- **Statement B'1:** The undirected graph D^{und} has a proper 2-coloring.

- **Statement B′2:** The digraph D has no cycles of odd length.
- **Statement B′3:** The digraph D has no circuits of odd length.

Proof. See §6.2.2 in the notes. □

5.3. The Brooks theorems

Criteria for the existence of a proper k-coloring exist – both sufficient and necessary ones, but not sufficient-and-necessary ones (unless $k = 2$ as we saw above, or $k = 1$ for obvious reasons). Here is a simple but nice sufficient criterion:

Theorem 5.3.1 (Little Brooks theorem)**.** Let $G = (V, E, \varphi)$ be a loopless multigraph with at least one vertex. Let

$$\alpha := \max \{\deg v \mid v \in V\}.$$

Then, G has a proper $(\alpha + 1)$-coloring.

Proof. Let $v_1, v_2, \ldots, v_n$ be the vertices of V, listed in some order. We construct a proper $(\alpha + 1)$-coloring $f : V \to \{1, 2, \ldots, \alpha + 1\}$ as follows:

- First, we choose $f(v_1)$ arbitrary.
- Then, we choose $f(v_2)$ to be distinct from the colors of all already-colored neighbors of v_2.
- Then, we choose $f(v_3)$ to be distinct from the colors of all already-colored neighbors of v_3.
- Then, we choose $f(v_4)$ to be distinct from the colors of all already-colored neighbors of v_4.
- And so on, until all colors $f(v_i)$ have been chosen.

In this process, we never run out of colors (since each vertex v_i satisfies $\deg v_i \leq \alpha$, which means that no more than α of the $\alpha + 1$ colors are forbidden for $f(v_i)$).

The resulting $(\alpha + 1)$-coloring is clearly proper. It is called a **greedy coloring**. □

In general, the $\alpha + 1$ in the Little Brooks theorem cannot be improved. Here are two examples:

- If $n \geq 1$, then the complete graph K_n has maximum degree $\alpha = n - 1$, and thus has a proper n-coloring by Little Brooks. And this is sharp, since it has no proper $(n-1)$-coloring.

- If $n \geq 2$ is odd, then the cycle graph C_n has maximum degree $\alpha = 2$, and thus has a proper 3-coloring by Little Brooks. And this is sharp, since it has no proper 2-coloring.

Interestingly, these are the only two cases in which the Little Brooks bound is sharp (unless G is disconnected)! In all other cases, we can improve the $\alpha + 1$ to α:

Theorem 5.3.2 (Brooks theorem)**.** Let $G = (V, E, \varphi)$ be a connected loopless multigraph. Let

$$\alpha := \max \{\deg v \mid v \in V\}.$$

Assume that G is neither a complete graph nor an odd-length cycle. Then, G has a proper α-coloring.

Proof. I give a reference in the notes (Theorem 6.3.2). □

5.4. The chromatic polynomial

Surprisingly, the # of proper k-colorings of a given graph G is a polynomial function in k. It can be described explicitly:

Theorem 5.4.1 (Whitney's chromatic polynomial theorem)**.** Let $G = (V, E, \varphi)$ be a multigraph. Define a univariate polynomial χ_G with integer coefficients as follows:

$$\chi_G(x) = \sum_{F \subseteq E} (-1)^{|F|} x^{\operatorname{conn}(V,F,\varphi|_F)} = \sum_{\substack{H \text{ is a spanning} \\ \text{subgraph of } G}} (-1)^{|\operatorname{E}(H)|} x^{\operatorname{conn} H}.$$

Then, for any $k \in \mathbb{N}$, we have

$$(\text{\# of proper } k\text{-colorings of } G) = \chi_G(k).$$

Very rough proof idea. Start with

$$\begin{aligned} &(\text{\# of proper } k\text{-colorings of } G) \\ &= \underbrace{(\text{\# of } k\text{-colorings of } G)}_{\substack{= k^n \\ (\text{where } n = |V|)}} - (\text{\# of improper } k\text{-colorings of } G). \end{aligned}$$

Now, improper k-colorings are k-colorings that leave some edge monochromatic (= both endpoints having the same color). So "to the first order of approximation",

$$\begin{aligned}
&(\text{\# of proper } k\text{-colorings of } G) \\
&= \underbrace{(\text{\# of } k\text{-colorings of } G)}_{=k^n} \\
&\qquad - \sum_{e \in E} \underbrace{(\text{\# of } k\text{-colorings of } G \text{ that leave } e \text{ monochromatic})}_{\substack{=k^{n-1} \text{ unless } e \text{ is a loop;} \\ \text{otherwise } k^n}} \\
&= k^n - \sum_{e \in E} \underbrace{\begin{cases} k^n, & \text{if } e \text{ is a loop;} \\ k^{n-1}, & \text{if not} \end{cases}}_{=k^{\operatorname{conn}\left(V, \{e\}, \varphi|_{\{e\}}\right)}}.
\end{aligned}$$

But now you have overcorrected: A k-coloring of G that leaves two edges e and f monochromatic is being subtracted twice. So you need to re-count it by adding back

$$\sum_{\substack{e,f \in E \\ \text{distinct}}} (\text{\# of } k\text{-colorings of } G \text{ that leave } e \text{ and } f \text{ monochromatic}).$$

This way, you end up with an inclusion/exclusion sum

$$\sum_{F \subseteq E} (-1)^{|F|} (\text{\# of } k\text{-colorings of } G \text{ that leave each edge from } F \text{ monochromatic}).$$

It remains to show that

$$\begin{aligned}
&(\text{\# of } k\text{-colorings of } G \text{ that leave each edge from } F \text{ monochromatic}) \\
&= k^{\operatorname{conn}(V, F, \varphi|_F)}
\end{aligned}$$

for each $F \subseteq E$. Details in the notes (§6.5). □

More in the notes.

6. Independent sets

6.1. Definition and lower bound

Definition 6.1.1. An **independent set** of a graph G means a subset of $\operatorname{V}(G)$ such that no two elements of S are adjacent.

In other words, an independent set of G means an induced subgraph of G that has no edges.

For example, if G is K_4 minus the edge 24, then the independent sets of G are $\varnothing$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$ and $\{2,4\}$.

Remark 6.1.2. Independent sets are closely to related to proper colorings.

Indeed, let G be a graph and $k \in \mathbb{N}$. Let $f : V \to \{1, 2, \ldots, k\}$ be any k-coloring. For each $i \in \{1, 2, \ldots, k\}$, let

$$V_i := \{v \in V \mid f(v) = i\}.$$

Then, f is a proper k-coloring if and only if the k sets $V_1, V_2, \ldots, V_k$ are independent sets of G.

An obvious computational problem is to find a maximum-size independent set of a given graph G. This problem is NP-hard, so don't expect a quick algorithm or a good formula for the size. However, there is a lower bound:

Theorem 6.1.3 (Caro–Wei theorem)**.** Let $G = (V, E, \varphi)$ be a loopless multigraph. Then, G has an independent set of size

$$\geq \sum_{v \in V} \frac{1}{1 + \deg v}.$$

First proof. Assume the contrary. Thus, each independent set of G has size

$$|S| < \sum_{v \in V} \frac{1}{1 + \deg v}.$$

A V**-listing** shall mean a list of all vertices in V (each occuring exactly once). If σ is a V-listing, then we define a subset J_σ of V as follows:

$$J_\sigma := \{v \in V \mid v \text{ occurs } \textbf{before} \text{ all neighbors of } v \text{ in } \sigma\}.$$

This subset J_σ is always an independent set of G (because if two vertices v and w in J_σ were adjacent, then the one of them that appears second in σ would not actually satisfy the requirement to go into J_σ). So, by our assumption, we get

$$|J_\sigma| < \sum_{v \in V} \frac{1}{1 + \deg v}.$$

This inequality holds for **each** V-listing σ. Summing it over all σ's, we obtain

$$\sum_{\sigma \text{ is a } V\text{-listing}} |J_\sigma| < (\text{\# of all } V\text{-listings}) \cdot \sum_{v \in V} \frac{1}{1 + \deg v}.$$

On the other hand:

Claim 1: For each $v \in V$, we have

$$(\text{\# of all } V\text{-listings } \sigma \text{ satisfy } v \in J_\sigma) \geq \frac{(\text{\# of all } V\text{-listings})}{1 + \deg v}.$$

Proof of Claim 1. For a V-listing σ to satisfy $v \in J_\sigma$, it must contain v before all the neighbors of v. The probability for this (if we choose σ uniformly) is

$$\frac{1}{1 + (\text{\# of neighbors of } v)} \geq \frac{1}{1 + \deg v}$$

(the $\geq$ sign is because we may have parallel edges). □

Now,

$$\begin{aligned}
&(\text{\# of all } V\text{-listings}) \cdot \sum_{v \in V} \frac{1}{1 + \deg v} \\
&> \sum_{\sigma \text{ is a } V\text{-listing}} |J_\sigma| \\
&= \sum_{\sigma \text{ is a } V\text{-listing}} (\text{\# of vertices } v \in V \text{ such that } v \in J_\sigma) \\
&= (\text{\# of pairs } (v, \sigma) \text{ where } v \in V \text{ and } \sigma \text{ is a } V\text{-listing and } v \in J_\sigma) \\
&= \sum_{v \in V} \underbrace{(\text{\# of } V\text{-listings } \sigma \text{ such that } v \in J_\sigma)}_{\substack{\geq \frac{(\text{\# of all } V\text{-listings})}{1 + \deg v} \\ \text{(by Claim 1)}}} \\
&\geq \sum_{v \in V} \frac{(\text{\# of all } V\text{-listings})}{1 + \deg v} \\
&= (\text{\# of all } V\text{-listings}) \cdot \sum_{v \in V} \frac{1}{1 + \deg v}.
\end{aligned}$$

Contradiction! Thus the proof is complete. □

Remark 6.1.4. This proof is an example of a **probabilistic proof**. In fact, our above manipulations of sums can be easily recast as manipulations of probabilities and expectations. It does not give a good algorithm for **finding** an independent set of size $\geq \sum_{v \in V} \frac{1}{1 + \deg v}$. The following alternative proof does, however:

Second proof. We proceed by strong induction on $|V|$. Thus, we fix $p \in \mathbb{N}$, and we assume (as IH) that the theorem holds for all loopless multigraphs G with $< p$ vertices. We must now prove it for a loopless multigraph $G = (V, E, \varphi)$ with p vertices.

If $|V| = 0$, then this is clear. So WLOG $|V| \neq 0$, so that $p \neq 0$.

WLOG G is a simple graph (since we can replace G by G^{simp} and the bound only gets sharper).

Pick a vertex $u \in V$ with $\deg_G u$ minimum.

Let $U := \{u\} \cup \{\text{all neighbors of } u\}$. Thus, $U \subseteq V$ and $|U| = 1 + \deg_G u$.

Let G' be the induced subgraph $G[V \setminus U]$. Hence, G' has $< p$ vertices, so the IH shows that it has an independent set of size

$$\geq \sum_{v \in V \setminus U} \frac{1}{1 + \deg_{G'} v}.$$

Let T be such an independent set, and let $S := \{u\} \cup T$. Then, S is an independent set of G. Now it remains to prove that $|S| \geq \sum\limits_{v \in V} \frac{1}{1 + \deg_G v}$.

Indeed, this follows from

$$\begin{aligned}
\sum_{v \in V} \frac{1}{1 + \deg_G v} &= \sum_{v \in U} \underbrace{\frac{1}{1 + \deg_G v}}_{\substack{\leq \frac{1}{1 + \deg_G u} \\ \text{(since } u \text{ has minimum degree,} \\ \text{so } \deg_G u \leq \deg_G v)}} + \sum_{v \in V \setminus U} \underbrace{\frac{1}{1 + \deg_G v}}_{\leq \frac{1}{1 + \deg_{G'} v}} \\
&\leq \underbrace{\sum_{v \in U} \frac{1}{1 + \deg_G u}}_{\substack{= |U| \cdot \frac{1}{1 + \deg_G u} = 1 \\ \text{(since } |U| = 1 + \deg_G u)}} + \underbrace{\sum_{v \in V \setminus U} \frac{1}{1 + \deg_{G'} v}}_{\leq |T|} \\
&\leq 1 + |T| = |S| \qquad (\text{since } S = \{u\} \cup T),
\end{aligned}$$

qed. □

Remark 6.1.5. This second proof (unlike the first one) gives a fairly efficient algorithm for finding an independent set of size $\geq \sum\limits_{v \in V} \frac{1}{1 + \deg_G v}$.

6.2. A weaker (but simpler) lower bound

Corollary 6.2.1. Let G be a loopless multigraph with n vertices and m edges. Then, G has an independent set of size

$$\geq \frac{n^2}{n + 2m}.$$

Proof. This is a weakening of the Caro–Wei theorem, since we have

$$\sum_{v \in V} \frac{1}{1 + \deg v} \geq \frac{n^2}{n + 2m}.$$

Why do we have this? This is a particular case of the following inequality:

$$\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_n} \geq \frac{n^2}{a_1 + a_2 + \cdots + a_n}$$

(for all $a_1, a_2, \ldots, a_n \in \mathbb{R}_{>0}$). This inequality can, in turn, be proved in many ways:

- Jensen (convex function $x \mapsto \frac{1}{x}$).
- Cauchy–Schwarz for $(a_1 + a_2 + \cdots + a_n) \left(\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_n} \right)$.
- AM-GM in the form $\frac{u}{v} + \frac{v}{u} \geq 2$.
- AM-HM.
- AM-GM twice.

□

This corollary, while weak, actually yields Turan's theorem:

Theorem 6.2.2 (Turan's theorem)**.** Let r be a positive integer. Let G be a simple graph with n vertices and e edges, where $e > \frac{r-1}{r} \cdot \frac{n^2}{2}$. Then, G has $r+1$ mutually adjacent vertices (i.e., every two distinct ones among them are adjacent).

Proof. Write G as $G = (V, E)$. Thus, $|V| = n$ and $|E| = e$.

Let $E' := \mathcal{P}_2(V) \setminus E$; thus, E' consists of all "non-edges" of G. Of course, $|E'| = \binom{n}{2} - e$.

Now, let G' be the simple graph (V, E'); this is called the **complementary graph** of G. It has n vertices and $\binom{n}{2} - e$ edges. So the preceding corollary shows that it has an independent set of size

$$\geq \frac{n^2}{n + 2\left(\binom{n}{2} - e\right)} = \frac{n^2}{n + n(n-1) - 2e} = \frac{n^2}{n^2 - 2e} > r$$

because $e > \frac{r-1}{r} \cdot \frac{n^2}{2}$. So G' has an independent set of size $\geq r+1$. This means that G has a complete subgraph of the form K_{r+1}, meaning $r+1$ vertices with all possible adjacencies. This proves Turan's theorem. □

7. Matchings

7.1. Introduction

Independent sets of a graph are sets of vertices that contain no two adjacent vertices.

"Dually", matchings of a graph are sets of edges that contain no two edges with a common endpoint. Here is the formal definition:

Definition 7.1.1. Let $G = (V, E, \varphi)$ be a loopless multigraph.

(a) A **matching** of G means a subset M of E such that no two distinct edges in M have a common endpoint.

(b) If M is a matching of G, then an M**-edge** means an edge belonging to M.

(c) If M is a matching of G, and if $v \in V$ is any vertex, then we say that v is **matched** (or **saturated**) in M if v is an endpoint of an M-edge. In this case, the latter M-edge is unique, and is called the M**-edge** of v. The other endpoint of this edge is called the M**-partner** of v.

(d) A matching M of G is said to be **perfect** if each vertex of G is matched in M.

(e) Let A be a subset of V. A matching M of G is said to be A**-complete** if each vertex in A is matched in M.

(So "perfect" = "V-complete".)

So a matching "pairs up" some vertices using the existing edges of the graph. Clearly, the M-partner of the M-partner of a vertex v is v itself. No two distinct vertices have the same M-partner.

Remark 7.1.2. We can also describe a matching of a loopless multigraph $G = (V, E, \varphi)$ as a spanning subgraph in which all vertices have degree ≤ 1.

Here are some natural questions:

- Does a given graph G have a perfect matching?
- If not, can we find a maximum-size matching?
- What about an A-complete matching for a given $A \subseteq V$?

Bad news: Greedy algorithms don't work.

Good news: There are polynomial-time algorithms (Edmonds blossom algorithm with running time $O\left(|E| \cdot |V|^2\right)$).

Bad news: These are still too complicated to be covered in this course.

But we will deal with one particularly useful case of the problem, which is when the graph is a bipartite graph.

7.2. Bipartite graphs

Definition 7.2.1. A **bipartite graph** means a triple (G, X, Y), where

- $G = (V, E, \varphi)$ be a multigraph, and
- X and Y are two disjoint subsets of V such that $X \cup Y = V$ and such that each edge of G has one endpoint in X and one endpoint in Y.

Example 7.2.2. Consider the 6-th cycle graph C_6. Then,

$$(C_6,\ \{1,3,5\},\ \{2,4,6\})$$

is a bipartite graph. Also,

$$(C_6,\ \{2,4,6\},\ \{1,3,5\})$$

is a bipartite graph.

We typically draw a bipartite graph (G, X, Y) by drawing the graph G in such a way that the X-vertices are aligned on a vertical line on the left, and the Y-vertices are aligned on a vertical line on the right.

Definition 7.2.3. Let (G, X, Y) be a bipartite graph. We shall refer to the vertices in X as the **left vertices** of this bipartite graph. We shall refer to the vertices in Y as the **right vertices** of this bipartite graph. We shall refer to the edges of G as the **edges** of this bipartite graph.

So each edge of a bipartite graph joins one left vertex and one right vertex.

Bipartite graphs are "the same as" multigraphs with a proper 2-coloring. To wit:

Proposition 7.2.4. Let $G = (V, E, \varphi)$ be a multigraph.

(a) If (G, X, Y) is a bipartite graph, then the map

$$f : V \to \{1,2\},$$
$$v \mapsto \begin{cases} 1, & \text{if } v \in X; \\ 2, & \text{if } v \in Y \end{cases}$$

is a proper 2-coloring of G.

(b) Conversely, if $f : V \to \{1,2\}$ is a proper 2-coloring of G, then $\left(G, f^{-1}(1), f^{-1}(2)\right)$ is a bipartite graph.

(c) These constructions are mutually inverse.

Corollary 7.2.5. Let (G, X, Y) be a bipartite graph. Then, G has no circuits of odd length. In particular, G has no loops or triangles.

One more piece of notation:

Definition 7.2.6. Let $G = (V, E, \varphi)$ be any multigraph. Let U be a subset of V. Then, set

$$N(U) := \{v \in V \mid v \text{ has a neighbor in } U\}.$$

This is called the **neighbor set** of U.

For bipartite graphs, if U is on one side, then $N(U)$ is on the opposite side:

Proposition 7.2.7. Let (G, X, Y) be a bipartite graph. Let $A \subseteq X$. Then, $N(A) \subseteq Y$.

7.3. Hall's marriage theorem

How can we tell whether a bipartite graph has a perfect matching? an X-complete matching? Let us first prove some trivialities:

Proposition 7.3.1. Let (G, X, Y) be a bipartite graph. Let M be a matching of G. Then:

(a) The M-partner of a vertex $x \in X$ (if it exists) belongs to Y.

The M-partner of a vertex $y \in Y$ (if it exists) belongs to X.

(b) We have $|M| \leq |X|$ and $|M| \leq |Y|$.

(c) If M is X-complete, then $|X| \leq |Y|$.

(d) If M is perfect, then $|X| = |Y|$.

Proof. Easy. □

The next (equally obvious) fact gives a necessary condition for the existence of an X-complete matching:

Proposition 7.3.2. Let (G, X, Y) be a bipartite graph. Let A be a subset of X. Assume that G has an X-complete matching. Then, $|N(A)| \geq |A|$.

Proof. Let V be the vertex set of G. Let M be an X-complete matching of G (this exists by assumption). Then, the map

$$\begin{aligned} \mathbf{p} : X &\to V, \\ x &\mapsto (\text{the } M\text{-partner of } x) \end{aligned}$$

is injective. Thus, $|\mathbf{p}(A)| = |A|$. But $\mathbf{p}(A) \subseteq N(A)$ (since a partner is a neighbor). So $|\mathbf{p}(A)| \leq |N(A)|$, thus $|N(A)| \geq |\mathbf{p}(A)| = |A|$. $\square$

Interestingly, this necessary condition is also sufficient:

Theorem 7.3.3 (Hall's marriage theorem, short HMT)**.** Let (G, X, Y) be a bipartite graph. Assume that each subset A of X satisfies $|N(A)| \geq |A|$. (This is called the **Hall condition**.)

Then, G has an X-complete matching.

This theorem was originally found by Philip Hall in 1935, as well as by Wilhelm Maak in 1935. By now there are myriad proofs of the theorem, including several elementary and self-contained ones that take no longer than a page (references in the notes). I will prove it using the theory of **network flows**, which arose from operations research in the 1950s, and generalizes the problem of finding a maximum-size matching for a bipartite graph. This theory also provides a polynomial-time algorithm for computing such a matching.

Before we get to this theory, however, let us introduce some related concepts.

7.4. König and Hall–König

The HMT is famous for its many versions and variants, most of which are "secretly" equivalent to it. We will start with one know as **König's theorem** (discovered independently by König and Egerváry in 1931). This requires the following notion:

Definition 7.4.1. Let $G = (V, E, \varphi)$ be a multigraph. A **vertex cover** of G means a subset C of V such that each edge of G contains at least one vertex in C.

Here is a little table of related notions:

a ...	**is a set of ...**	**that contains ...**
matching	edges	at most one edge per vertex
edge cover	edges	at least one edge per vertex
independent set	vertices	at most one vertex per edge
vertex cover	vertices	at least one vertex per edge

Note that each vertex cover of G is a dominating set if G has no isolated vertices. But the converse is not true.

Proposition 7.4.2 (easy)**.** Let G be a loopless multigraph.
Let m be the largest size of a matching of G.
Let c be the smallest size of a vertex cover of G.
Then, $m \leq c$.

Proof. Let M be a matching of size m. Let C be a vertex cover of size c. Thus, we can define a map $f : M \to C$ that sends each M-edge e to an endpoint of e that belongs to C (such an endpoint exists since C is a vertex cover). This map f is injective, since two M-edges cannot have an endpoint in common. So the pigeonhole principle yields $|M| \leq |C|$, that is, $m \leq c$. □

In general, we can have $m < c$ in this proposition. However, for **bipartite graphs**, equality holds:

Theorem 7.4.3 (König's theorem)**.** If (G, X, Y) is a bipartite graph, then in the proposition above we have $m = c$.

Both Hall's and König's theorem follow from the following theorem:

Theorem 7.4.4 (Hall–König matching theorem, short HKMT)**.** Let (G, X, Y) be a bipartite graph. Then, there exist a matching M of G and a subset U of X such that

$$|M| \geq |N(U)| + |X| - |U|.$$

In fact:

- To derive Hall's marriage theorem from the HKMT, you just observe that $|M| \geq \underbrace{|N(U)|}_{\geq |U|} + |X| - |U| \geq |X|$, which entails that M is X-complete.
- To derive König's theorem from the HKMT, you note that $(X \setminus U) \cup N(U)$ is a vertex cover of G that has size $\leq |M|$, so that you get $c \leq m$, and combining this with $m \leq c$, you conclude that $m = c$.

See the notes for details.

Next time, we will prove the HKMT using network flows, and we will see some applications of the HMT.

8. Networks and flows

Today: a short introduction to **network flows** and their optimization. We will prove one of the main theorems called the max-flow-min-cut theorem (for integer-valued flows), and then derive the HKMT from it.

8.1. Definition

Recall that $\mathbb{N} = \{0, 1, 2, \ldots\}$.

Definition 8.1.1. A **network** consists of

- a multidigraph $D = (V, A, \psi)$;
- two distinct vertices $s, t \in V$, called the **source** and the **sink**, respectively;
- a function $c : A \to \mathbb{N}$, called the **capacity function**. Its values $c(a)$ are called the **capacities** of the respective arcs a.

We do **not** require that $\deg^- s = 0$ or $\deg^+ t = 0$. We allow some capacities $c(a)$ to be 0, but an arc with capacity 0 will not contribute anything of use.

Definition 8.1.2. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source s, a sink t and a capacity function $c : A \to \mathbb{N}$. Then:

1. For any $S \subseteq V$, we let $\overline{S} := V \setminus S$.
2. If P and Q are two subsets of V, then $[P, Q]$ shall mean the set of all arcs of D whose source belongs to P and whose target belongs to Q. That is,
$$[P, Q] := \{a \in A \mid \psi(a) \in P \times Q\}.$$
3. If P and Q are two subsets of V, and if $d : A \to \mathbb{N}$ is any function, then we define
$$d(P, Q) := \sum_{a \in [P,Q]} d(a) \in \mathbb{N}.$$
In particular,
$$c(P, Q) := \sum_{a \in [P,Q]} c(a) \in \mathbb{N}.$$
4. A **flow** (on the network N) means a function $f : A \to \mathbb{N}$ with the following properties:
   a) We have $0 \leq f(a) \leq c(a)$ for each arc $a \in A$. This condition is called the **capacity constraints**.
   b) For any vertex $v \in V \setminus \{s, t\}$, we have
   $$f^-(v) = f^+(v),$$
   where we set
   $$f^-(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with target } v}} f(a)}_{\text{"}\textbf{inflow}\text{ into } v\text{"}} \qquad \text{and} \qquad f^+(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with source } v}} f(a)}_{\text{"}\textbf{outflow}\text{ from } v\text{"}}.$$

This is called the **conservation constraints**.

There are several ways to think of a network N and a flow on it:

- N is a collection of (one-way) roads. $c(a)$ is how much traffic road a can handle (per hour). $f(a)$ is how much traffic actually flows through a in a given hour. The conservation constraints say that the traffic out of a given vertex v equals the traffic into v unless v is one of s and t.
- N is a collection of pipes. $c(a)$ is how much water a pipe can carry (per hour). $f(a)$ is the actual amount of water flowing through a in a given hour. Water is produced at s and consumed at t, but all other nodes v are meant to be neutral (inflow = outflow).
- N is a money transfer scheme, where s is transferring money to t via a bunch of middlemen (the other vertices).

Remark 8.1.3. Flows on a network N can be viewed as generalizations of paths and of cycles on the corresponding digraph D. Indeed, if $\mathbf{p}$ is a path from s to t on the digraph $D = (V, A, \psi)$ underlying a network N, then we can define a flow $f_{\mathbf{p}}$ on N as follows:

$$f_{\mathbf{p}}(a) = \begin{cases} 1, & \text{if } a \text{ is an arc of } \mathbf{p}; \\ 0, & \text{otherwise} \end{cases} \qquad \text{for each } a \in A,$$

provided that all arcs of $\mathbf{p}$ have capacity ≥ 1. Likewise for cycles.

Definition 8.1.4. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source s, a sink t and a capacity function $c : A \to \mathbb{N}$. Let $f : A \to \mathbb{N}$ be any map. Then:

1. For any vertex $v \in V \setminus \{s, t\}$, we set

$$f^{-}(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with target } v}} f(a)}_{\text{"}\textbf{inflow}\text{ into } v\text{"}} \qquad \text{and} \qquad f^{+}(v) := \underbrace{\sum_{\substack{a \in A \text{ is an arc} \\ \text{with source } v}} f(a)}_{\text{"}\textbf{outflow}\text{ from } v\text{"}}.$$

2. We define the **value** $|f|$ of the map f to be the number $f^{+}(s) - f^{-}(s)$.

Now we can state an important optimization problem, known as the **maximum flow problem**: Given a network N, how we can we find a flow of maximum possible value?

Example 8.1.5. Finding a maximum matching in a bipartite graph (G, X, Y) is a particular case of the maximum flow problem. Here, the network consists of the graph G with each edge oriented from X to Y and with two extra vertices s and t adjoined and connected to respectively the X-vertices and the Y-vertices. Each arc has capacity 1.

8.2. Basic properties of flows

Proposition 8.2.1. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source s, a sink t and a capacity function $c : A \to \mathbb{N}$. Let $f : A \to \mathbb{N}$ be any flow on N. Then,

$$\begin{aligned} |f| &= f^+(s) - f^-(s) \\ &= f^-(t) - f^+(t). \end{aligned}$$

Proof. We have

$$\sum_{v \in V} f^+(v) = \sum_{a \in A} f(a) = \sum_{v \in V} f^-(v),$$

so

$$\sum_{v \in V} \left(f^+(v) - f^-(v)\right) = 0.$$

But the sum on the left has only two nonzero addends: that for $v = s$ and that for $v = t$ (since each $v \notin \{s, t\}$ satisfies the conservation constraint $f^+(v) = f^-(v)$). Thus, the equality simplifies to

$$\left(f^+(s) - f^-(s)\right) + \left(f^+(t) - f^-(t)\right) = 0,$$

that is,

$$f^+(s) - f^-(s) = f^-(t) - f^+(t),$$

qed. □

Proposition 8.2.2. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source s, a sink t and a capacity function $c : A \to \mathbb{N}$. Let $f : A \to \mathbb{N}$ be any flow on N. Let S be a subset of V. Then:

1. We have

$$f\left(S, \overline{S}\right) - f\left(\overline{S}, S\right) = \sum_{v \in S} \left(f^+(v) - f^-(v)\right).$$

2. Assume that $s \in S$ and $t \notin S$. Then,
$$|f| = f\left(S, \overline{S}\right) - f\left(\overline{S}, S\right).$$

3. Assume that $s \in S$ and $t \notin S$. Then,
$$|f| \leq c\left(S, \overline{S}\right).$$

4. Assume that $s \in S$ and $t \notin S$. Then, $|f| = c\left(S, \overline{S}\right)$ if and only if
$$\left(f(a) = 0 \text{ for all } a \in \left[\overline{S}, S\right]\right)$$
and
$$\left(f(a) = c(a) \text{ for all } a \in \left[S, \overline{S}\right]\right).$$

Proof. Easy and omitted (see notes). □

8.3. The max-flow-min-cut theorem

Theorem 8.3.1 (max-flow-min-cut theorem)**.** Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source s, a sink t and a capacity function $c : A \to \mathbb{N}$. Then,
$$\max\{|f| \mid f \text{ is a flow}\} = \min\left\{c\left(S, \overline{S}\right) \mid S \subseteq V;\ s \in S;\ t \notin S\right\}.$$
In other words, the maximum value of a flow equals the minimum capacity of a cut. Here, a **cut** means a subset of A that has the form $\left[S, \overline{S}\right]$ for some $S \subseteq V$ satisfying $s \in S$ and $t \notin S$; and the **capacity** of such a cut is defined to be $c\left(S, \overline{S}\right)$.

Next time we will prove this theorem. And we will do this algorithmically, using the **Ford–Fulkerson algorithm**, which constructs both a maximum-value flow and a minimum-capacity cut.

The algorithm will be incremental: We start with f being the zero flow ($f(a) = 0$ for all $a \in A$), and gradually increase its value $|f|$ until we reach $|f| = c\left(S, \overline{S}\right)$ for some cut $\left(S, \overline{S}\right)$.

We can do this using some form of "zig-zag paths" which use some arcs of D in the forward direction and some in the backward directions. The easiest way to formalize this is by defining a new digraph, on which these "zig-zag paths" will just become regular paths:

Definition 8.3.2. Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$.

(a) For each arc $a \in A$, we introduce a new arc a^{-1}, which should act as a reversal of a (so its source is the target of a and vice versa). We also set $\left(a^{-1}\right)^{-1} := a$.

We shall refer to the arcs $a \in A$ as **forward arcs**, and to their reversals a^{-1} as **backwards arcs**.

(b) Let $f : A \to \mathbb{N}$ be a flow on N. We define the **residual digraph** D_f of this flow f to be the multidigraph $\left(V, A_f, \psi_f\right)$, where

$$A_f = \{a \in A \mid f(a) < c(a)\} \cup \left\{a^{-1} \mid a \in A \text{ and } f(a) > 0\right\},$$

and where ψ_f does what it should (meaning $\psi_f(a) = \psi(a)$ and $\psi_f\left(a^{-1}\right) = \operatorname{flip}(\psi(a))$).

Now the "zig-zag paths" from s to t that allow us to change the flow are just the usual paths from s to t on the residual digraph. Let us state this as a lemma:

Lemma 8.3.3 (augmenting path lemma)**.** Let N be a network consisting of a multidigraph $D = (V, A, \psi)$, a source $s \in V$, a sink $t \in V$ and a capacity function $c : A \to \mathbb{N}$.

Let $f : A \to \mathbb{N}$ be a flow.

(a) If the digraph D_f has a path from s to t, then the network N has a flow f' with a larger value than f. Such a flow f' can be obtained from f by incrementing (by 1) the values $f(a)$ for all arcs $a \in A$ with a lying on this path, and decrementing (by 1) the values $f(a)$ for all arcs $a \in A$ with a^{-1} lying on this path.

(b) If the digraph D_f has no path from s to t, then the flow f has a maximum value, and there exists a subset S of V satisfying $s \in S$ and $t \notin S$ and $|f| = c\left(S, \overline{S}\right)$.

Proof. See notes.

(a) just requires a direct verification that the procedure mentioned does produce a flow.

(b) This set S is the set of all vertices $v \in V$ such that D_f has a path from s to v. To prove that $|f| = c\left(S, \overline{S}\right)$, we must show that

$$\left(f(a) = 0 \text{ for all } a \in \left[\overline{S}, S\right]\right)$$

and

$$\left(f(a) = c(a) \text{ for all } a \in \left[S, \overline{S}\right]\right).$$

But this is obvious (since the definition of S shows that D_f has no arc with source in S and target in $\overline{S}$, and we just need to recall how the arcs of D_f were defined). $\square$

Proof the max-flow-min-cut theorem. We start with zero flow $f : A \to \mathbb{N}$, and we incrementally augment it (i.e., increase its value) by finding a path from s to t in the residual digraph D_f and proceeding as in part **(a)** of the augmenting path lemma. Part **(b)** of that lemma shows that once such path no longer exists, we are done (i.e., our flow f is maximal). This is called the **Ford–Fulkerson algorithm**.

Why does this algorithm terminate? Because the flow value $|f|$ increases by 1 at each step, and thus will eventually outgrow the capacity bound $c\left(S, \overline{S}\right)$ for any fixed cut $\left[S, \overline{S}\right]$. □

Remark 8.3.4. The max-flow-min-cut theorem is true even if we replace $\mathbb{N}$ by $\mathbb{Q}_+$ or by $\mathbb{R}_+$. However, our proof no longer works, since we can increment a rational or real number indefinitely without ever surpassing a given bound like $c\left(S, \overline{S}\right)$.

The case of $\mathbb{Q}_+$ can be reduced to the case of $\mathbb{N}$ by multiplying all $c(a)$ values with their lowest common denominator.

The case of $\mathbb{R}_+$ requires new ideas. One way to do this is using the **Edmonds–Karp version of the Ford–Fulkerson algorithm**, in which you always pick a shortest path from s to t in D_f. Proving that this actually works is still not obvious. This also ensures that the algorithm runs in running time $O\left(|V| \cdot |A|^2\right)$.

8.4. Application: Deriving Hall–König

Now we shall derive the HKMT from the max-flow-min-cut theorem:

Proof of HKMT.. We turn G into a directed graph by orienting each edge to go from the X-endpoint to the Y-endpoint. We then insert two new vertices s and t as well an arc from s to every X-vertex and an arc from every Y-vertex to t. We assign to every arc a of this digraph the capacity 1. Thus we obtain a network N. Recall that the flows on this network are in bijection with the matchings of G, and the value of a flow is the size of the corresponding matching.

The max-flow-min-cut theorem yields that

$$\max\{|f| \mid f \text{ is a flow}\} = \min\left\{c\left(S, \overline{S}\right) \mid S \subseteq V;\ s \in S;\ t \notin S\right\},$$

where V is the vertex set of our network. Thus, there exist a flow f and a subset $S \subseteq V$ with $s \in S$ and $t \notin S$ satisfy $|f| = c\left(S, \overline{S}\right)$.

Let M be the matching of G corresponding to the flow f. So we have $|M| = |f| = c\left(S, \overline{S}\right)$.

Let U be the set $X \cap S$. We shall show that $c\left(S, \overline{S}\right) \geq |N(U)| + |X| - |U|$.

Since all our arcs have capacity 1, the number $c\left(S,\overline{S}\right)$ is just the # of arcs in $\left[S,\overline{S}\right]$. Thus,

$$\begin{aligned}
c\left(S,\overline{S}\right) &= \underbrace{c\left(\{s\},\overline{S}\right)}_{=|X\setminus U|} + c\left(\underbrace{X\cap S}_{=U},\ \overline{S}\right) + \underbrace{c\left(Y\cap S,\ \overline{S}\right)}_{=|Y\cap S|} \\
&= \underbrace{|X\setminus U|}_{=|X|-|U|} + \underbrace{c\left(U,\overline{S}\right)+|Y\cap S|}_{\substack{\geq|N(U)| \\ \text{(since each vertex } y\in N(U) \text{ either belongs to } Y\cap S \\ \text{and thus contributes to } |Y\cap S|\text{, or belongs to } \overline{S} \\ \text{and so contributes to } c(U,\overline{S}))}} \\
&\geq |X|-|U|+|N(U)| = |N(U)|+|X|-|U|.
\end{aligned}$$

Hence,

$$|M| = |f| = c\left(S,\overline{S}\right) \geq |N(U)|+|X|-|U|.$$

□

8.5. Applications of Hall's Marriage Theorem

Theorem 8.5.1 (existence of SDR)**.** Let $A_1, A_2, \ldots, A_n$ be any n sets. Assume that the union of any p of these sets has size $\geq p$, for all $p \in \{0, 1, \ldots, n\}$. Then, we can find n **distinct** elements

$$a_1 \in A_1, \qquad a_2 \in A_2, \qquad \ldots, \qquad a_n \in A_n.$$

(This is called a **system of distinct representatives**.)

Theorem 8.5.2 (Frobenius matching theorem)**.** Let $k > 0$. Let (G, X, Y) be a k-regular bipartite graph – i.e., a bipartite graph such that each vertex of G has degree k. Then, G has a perfect matching.

Theorem 8.5.3 (Birkhoff–von Neumann theorem)**.** Any doubly stochastic $n \times n$-matrix is a convex combination of permutation matrices.

There is much more – see the notes.