An introduction to graph theory

(Text for Math 530 in Spring 2022 at Drexel University)

Darij Grinberg*

Spring 2023 edition, August 2, 2023

Abstract. This is a graduate-level introduction to graph theory, corresponding to a quarter-long course. It covers simple graphs, multigraphs as well as their directed analogues, and more restrictive classes such as tournaments, trees and arborescences. Among the features discussed are Eulerian circuits, Hamiltonian cycles, spanning trees, the matrix-tree and BEST theorems, proper colorings, Turan's theorem, bipartite matching and the Menger and Gallai–Milgram theorems. The basics of network flows are introduced in order to prove Hall's marriage theorem.

Around a hundred exercises are included (without solutions).

Contents

*Drexel University, Korman Center, 15 S 33rd Street, Office #263, Philadelphia, PA 19104 (USA). // darijgrinberg@gmail.com // http://www.cip.ifi.lmu.de/~grinberg/

1. Preface

1.1. What is this?

This is a course on **graphs** – a rather elementary concept (actually a cluster of closely related concepts) that can be seen all over mathematics. We will discuss several kinds of graphs (simple graphs, multigraphs, directed graphs, etc.) and study their features and properties. In particular, we will encounter walks on graphs, matchings of graphs, flows on networks (networks are graphs with extra data), and take a closer look at certain types of graphs such as trees and tournaments.

The theory of graphs goes back at least to Leonhard Euler, who in a 1736 paper [Euler36] (see [Euler53] for an English translation) solved a puzzle about an optimal tour of the town of Königsberg. It saw some more developments in the 19th century and straight-up exploded in the 20th; now it is one of the most active fields of mathematics. There are now dozens (if not hundreds) textbooks available on the subject, such as

- the comprehensive works [BonMur08], [Berge91], [Ore74], [Bollob98], [Dieste17], [ChLeZh16], [Jungni13]
- or the more introductory [Ore96], [BenWil06, Chapters 5–6], [Bollob71], [Griffi21], [Galvin21], [Guicha16, Chapter 5], [Harary69], [Harju14], [HaHiMo08, Chapter 1], [Wilson10], [Tait21], [LeLeMe18, Chapters 10–13], [Ruohon13], [KelTro17], [LoPeVe03], [West01], [Verstr21], [HarRin03].

These texts are written at different levels of sophistication, rigor and detail, are tailored to different audiences, and (beyond the absolute basics) often cover different ground (for instance, [Dieste17] distinguishes itself by treating infinite and random graphs, whereas [Griffi21] is strong on applications).

The present notes are self-contained and do not follow any existing book. Nevertheless, I recommend skimming the texts cited above to gain a wider perspective on graph theory (far beyond what we can cover in an introductory course), and perhaps marking the one or the other book for later reading. Our focus in these notes is on the more discrete and algebraic sides of graph theory (finite graphs of various kinds, existential results, counting formulas), and they are limited both by the time constraints (being written for a quarter-long course) and the limits of my own knowledge.

1.1.1. Remarks

Prerequisites. These notes target a graduate-level (or advanced undergraduate) reader. A certain mathematical sophistication and willingness to think along (as well as invent one's own examples) is expected. Beyond that, the main prerequisites are the basic properties of determinants, polynomials and finite sums. Rings and fields are occasionally mentioned, but the reader can make do with just the most basic examples thereof ($\mathbb{Q}$, $\mathbb{R}$, polynomial rings and matrix rings; also the finite field $\mathbb{F}_2$ in a few places). No analysis (or even calculus) is required anywhere in this text.

Course websites. These notes were written for my Math 530 course at Drexel University in Spring 2022. The website of this course can be found at

`https://www.cip.ifi.lmu.de/~grinberg/t/22s` .

An older, but similarly structured course is my Spring 2017 course at the University of Minnesota. Its website is available at

`https://www.cip.ifi.lmu.de/~grinberg/t/17s` ,

and contains some additional materials (such as solutions to some selected exercises, a few more detailed topics, and a stub of a text [17s] that covers parts of our Chapter 2 in more depth). If you are reading the present notes on the arXiv, then said additional materials can also be found as ancillary files to this arXiv submission.

Exercises. These notes include exercises of varying difficulty and significance. Almost all of the exercises are optional (i.e., they are not used anywhere in the text, except perhaps in other exercises), but they often provide practice, context and additional inspiration. Naturally, one person's inspiration is another's distraction, so I do not recommend assigning too much importance to any specific exercise; it is usually better to read on than to dwell for hours. However, a dozen minutes of thought per exercise will likely not be a waste of time.

Acknowledgments. I have learned a lot from conversations with Joel Brewster Lewis, Lukas Katthän and Victor Reiner. Chiara Libera Carnevale and Amanda Johnson corrected errors in previous versions of these notes. I am indebted to all of the above, and would appreciate any further input – please contact `darijgrinberg@gmail.com` about any corrections (however small) and suggestions.

1.2. Notations

The following notations will be used throughout these notes:

- We let $\mathbb{N} = \{0, 1, 2, \ldots\}$. Thus, $0 \in \mathbb{N}$.
- The size (i.e., cardinality) of a finite set S is denoted by $|S|$.
- If S is a set, then the **powerset** of S means the set of all subsets of S. This powerset will be denoted by $\mathcal{P}(S)$.

  Moreover, if S is a set, and k is an integer, then $\mathcal{P}_k(S)$ will mean the set of all k-element subsets of S. For instance,

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

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

  $$\frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} = \frac{\prod_{i=0}^{k-1}(n-i)}{k!}.$$

  These binomial coefficients have many interesting properties, which can often be found in textbooks on enumerative combinatorics (e.g., [19fco, Chapter 2]). Some of the most important ones are the following:

  - The factorial formula: If $n, k \in \mathbb{N}$ and $n \geq k$, then $\dbinom{n}{k} = \dfrac{n!}{k! \cdot (n-k)!}$.
  - The combinatorial interpretation: If $n, k \in \mathbb{N}$, and if S is an n-element set, then $\dbinom{n}{k}$ is the number of all k-element subsets of S (in other words, $|\mathcal{P}_k(S)| = \dbinom{n}{k}$).
  - Pascal's recursion: For any number n and any positive integer k, we have

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

2. Simple graphs

2.1. Definitions

The first type of graphs that we will consider are the "simple graphs", named so because of their very simple definition:

Definition 2.1.1. A **simple graph** is a pair (V, E), where V is a finite set, and where E is a subset of $\mathcal{P}_2(V)$.

To remind, $\mathcal{P}_2(V)$ is the set of all 2-element subsets of 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 the word "simple graph" as "graph" in this chapter, but later (in Chapter 3) we will learn some more advanced and general notions of "graphs".

Example 2.1.2. Here is a simple graph:

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

Example 2.1.3. For any $n \in \mathbb{N}$, we can define a simple graph Cop_n to be the pair (V, E), where $V = \{1, 2, \ldots, n\}$ and

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

We call this the **n-th coprimality graph**.

(Some authors do not require V to be finite in Definition 2.1.1; this leads to **infinite graphs**. But I shall leave this can of worms closed for this quarter.)

The purpose of simple graphs is to encode relations on a finite set – specifically the kind of relations that are binary (i.e., relate pairs of elements), symmetric (i.e., mutual) and irreflexive (i.e., an element cannot be related to itself). For example, the graph Cop_n in Example 2.1.3 encodes the coprimality (aka coprimeness) relation on the set $\{1, 2, \ldots, n\}$, except that the latter relation is not irreflexive (1 is coprime to 1, but $\{1, 1\}$ is not in E; thus, the graph Cop_n "forgets" that 1 is coprime to 1). For another example, if V is a set of people, and E is the set of $\{u, v\} \in \mathcal{P}_2(V)$ such that u has been married to v at some point, then (V, E) is a simple graph. Even in 2022, marriage to oneself is not a thing, so all marriages can be encoded as 2-element subsets.[1]

The following notations provide a quick way to reference the elements of V and E when given a graph (V, E):

[1]The more standard example for a social graph would be a "friendship graph"; here, V is again a set of people, but E is now the set of $\{u, v\} \in \mathcal{P}_2(V)$ such that u and v are friends. Of course, this only works if you think of friendship as being automatically mutual (true for facebook friendship, questionable for the actual thing).

Definition 2.1.4. Let $G = (V, E)$ be a simple graph.

(a) The set V is called the **vertex set** of G; it is denoted by $\operatorname{V}(G)$. (Notice that the letter "V" in "$\operatorname{V}(G)$" is upright, as opposed to the letter "V" in "(V, E)", which is italic. These are two different symbols, and have different meanings: The letter V stands for the specific set V which is the first component of the pair G, whereas the letter V is part of the notation $\operatorname{V}(G)$ for the vertex set of any graph. Thus, if $H = (W, F)$ is another graph, then $\operatorname{V}(H)$ is W, not V.)

The elements of V are called the **vertices** (or the **nodes**) of G.

(b) The set E is called the **edge set** of G; it is denoted by $\operatorname{E}(G)$. (Again, the letter "E" in "$\operatorname{E}(G)$" is upright, and stands for a different thing than the "E".)

The elements of E are called the **edges** of G. When u and v are two elements of V, we shall often use the notation uv for $\{u, v\}$; thus, each edge of G has the form uv for two distinct elements u and v of V. Of course, we always have $uv = vu$.

Notice 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 $uv \in E$ (that is, if uv is an edge of G). 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. When the graph G is not obvious from the context, we shall often say "adjacent in G" instead of just "adjacent".

Two vertices u and v of G are said to be **non-adjacent** (to each other) if they are not adjacent (i.e., if $uv \notin E$).

(d) Let v be a vertex of G (that is, $v \in V$). Then, the **neighbors** of v (in G) are the vertices u of G that satisfy $vu \in E$. In other words, the **neighbors** of v are the vertices of G that are adjacent to v.

Example 2.1.5. Let G be the simple graph

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

from Example 2.1.2. Then, its vertex set and its edge set are

$$\operatorname{V}(G) = \{1,2,3,4\} \quad \text{and} \quad \operatorname{E}(G) = \{\{1,3\}, \quad \{1,4\}, \quad \{3,4\}\} = \{13, \ 14, \ 34\}$$

(using our notation uv for $\{u, v\}$). The vertices 1 and 3 are adjacent (since $13 \in \operatorname{E}(G)$), but the vertices 1 and 2 are not (since $12 \notin \operatorname{E}(G)$). The neighbors of 1 are 3 and 4. The endpoints of the edge 34 are 3 and 4.

2.2. Drawing graphs

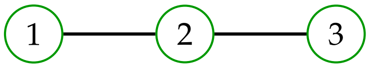
There is a common method to represent graphs visually: Namely, a graph can be drawn as a set of points in the plane and a set of curves connecting some of these points with each other.

More precisely:

Definition 2.2.1. 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 allow the reader to unambiguously reconstruct the graph G from the picture. (Thus, for example, the curves should not pass through any points other than the ones they mean to connect.)

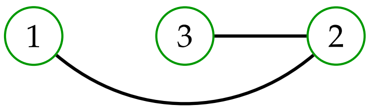
Example 2.2.2. Let us draw some simple graphs.

(a) The simple graph $(\{1,2,3\}, \{12,23\})$ (where we are again using the shorthand notation uv for $\{u,v\}$) can be drawn as follows:



.

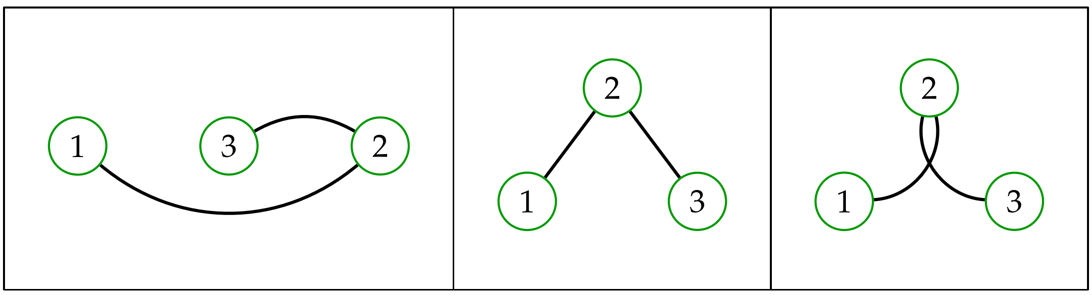
This is (in a sense) the simplest way to draw this graph: The edges are represented by straight lines. But we can draw it in several other ways as well – e.g., as follows:



.

Here, we have placed the points representing the vertices $1,2,3$ differently. As a consequence, we were not able to draw the edge 12 as a straight line, because it would then have overlapped with the vertex 3, which would make the graph ambiguous (the edge 12 could be mistaken for two edges 13 and 32).

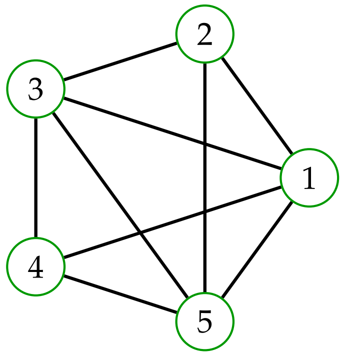
Here are three further drawings of the same graph $(\{1,2,3\}, \{12,23\})$:



.

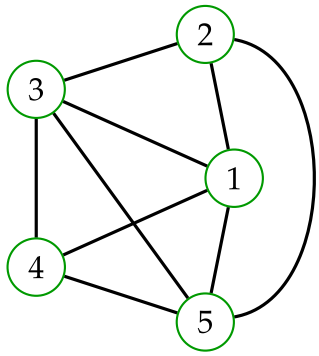
(b) Consider the 5-th coprimality graph Cop_5 defined in Example 2.1.3.

Here is one way to draw it:



.

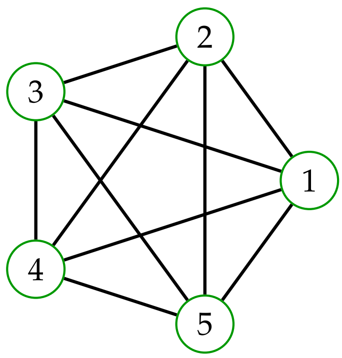
Here is another way to draw the same graph Cop_5, with fewer intersections between edges:



.

By appropriately repositioning the points corresponding to the five vertices of Cop_5, we can actually get rid of all intersections and make all the edges straight (as opposed to curved). Can you find out how?

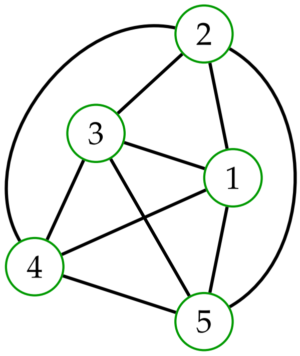
(c) Let us draw one further graph: the simple graph $(\{1,2,3,4,5\}, \mathcal{P}_2(\{1,2,3,4,5\}))$. This is the simple graph whose vertices are $1,2,3,4,5$, and whose edges are all possible two-element sets consisting of its vertices (i.e., each pair of two distinct vertices is adjacent). We shall later call this graph the "complete graph K_5". Here is a simple way to draw this graph:



.

This drawing is useful for many purposes; for example, it makes the abstract symmetry of this graph (i.e., the fact that, roughly speaking, its vertices

$1, 2, 3, 4, 5$ are "equal in rights") obvious. But sometimes, you might want to draw it differently, to minimize the number of intersecting curves. Here is a drawing with fewer intersections:



.

In this drawing, we have only one intersection between two curves left. Can we get rid of all intersections?

This is a question of topology, not of combinatorics, since it really is about curves in the plane rather than about finite sets and graphs. The answer is "no". (That is, no matter how you draw this graph in the plane, you will always have at least one pair of curves intersect.) This is a classical result (one of the first theorems in the theory of **planar graphs**), and proofs of it can be found in various textbooks (e.g., [FriFri98, Theorem 4.1.2], which is generally a good introduction to planar graph theory even if it uses terminology somewhat different from ours). Note that any proof must use some analysis or topology, since the result relies on the notion of a (continuous) curve in the plane (if curves were allowed to be non-continuous, then they could "jump over" one another, so they could easily avoid intersecting!).

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

Enough definitions; let's state a first result:

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

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

In other words, Proposition 2.3.1 says that if a graph G has at least 6 vertices, then we can either find three distinct vertices that are mutually adjacent[2] or find three distinct vertices that are mutually non-adjacent (i.e., no two of them are adjacent), or both. Often, this is restated as follows: "In any group of at least six people, you can always find three that are (pairwise) friends to each other, or three no two of whom are friends" (provided that friendship is a symmetric relation).

We will give some examples in a moment, but first let us introduce some convenient terminology:

Definition 2.3.2. Let G be a simple graph.

(a) A set $\{a, b, c\}$ of three distinct vertices of G is said to be a **triangle** (of G) if every two distinct vertices in this set are adjacent (i.e., 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** (of G) if no two distinct vertices in this set are adjacent (i.e., if none of ab, bc and ca is an edge of G).

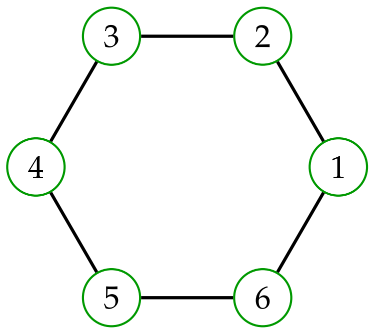
Thus, Proposition 2.3.1 says that every simple graph with at least 6 vertices contains a triangle or an anti-triangle (or both).

Example 2.3.3. Let us show two examples of graphs G to which Proposition 2.3.1 applies, as well as an example to which it does not:

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

$$V = \{1, 2, 3, 4, 5, 6\} \qquad \text{and}$$
$$E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 1\}\}.$$

(This graph can be drawn in such a way as to look like a hexagon:



.

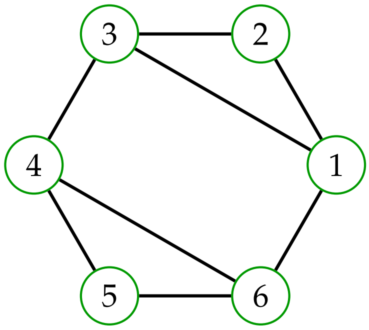
) This graph satisfies Proposition 2.3.1, since $\{1, 3, 5\}$ is an anti-triangle (or since $\{2, 4, 6\}$ is an anti-triangle).

[2]by which we mean (of course) that any two **distinct** ones among these three vertices are adjacent

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

$$V = \{1, 2, 3, 4, 5, 6\} \qquad \text{and}$$
$$E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 1\}, \{1, 3\}, \{4, 6\}\}.$$

(This graph can be drawn in such a way as to look like a hexagon with two extra diagonals:



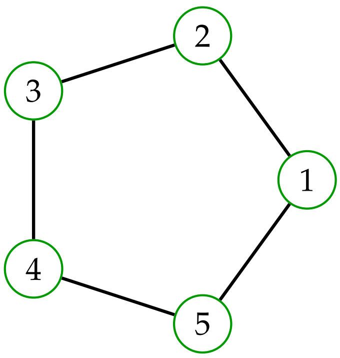
.

) This graph satisfies Proposition 2.3.1, since $\{1, 2, 3\}$ is a triangle.

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

$$V = \{1, 2, 3, 4, 5\} \qquad \text{and}$$
$$E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 1\}\}.$$

(This graph can be drawn to look like a pentagon:



.

) Proposition 2.3.1 says nothing about this graph, since this graph does not satisfy the assumption of Proposition 2.3.1 (in fact, its number of vertices $|\mathrm{V}(G)|$ fails to be ≥ 6). By itself, this does not yield that the claim of Proposition 2.3.1 is false for this graph. However, it is easy to check that the claim actually **is** false for this graph: It has neither a triangle nor an anti-triangle.

Proof of Proposition 2.3.1. We need to prove that G has a triangle or an anti-triangle (or both).

Choose any vertex $u \in \operatorname{V}(G)$. (This is clearly possible, since $|\operatorname{V}(G)| \geq 6 \geq 1$.) Then, there are at least 5 vertices distinct from u (since G has at least 6 vertices). 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.

Let us consider Case 1 first. In this case, the vertex u has at least 3 neighbors. Hence, we can find three distinct neighbors p, q and r of u. Consider these p, q and r. If one (or more) of pq, qr and rp is an edge of G, then G has a triangle (for example, if pq is an edge of G, then $\{u, p, q\}$ is a triangle). If not, then G has an anti-triangle (namely, $\{p, q, r\}$). Thus, in either case, our proof is complete in Case 1.

Let us now consider Case 2. In this case, the vertex u has at most 2 neighbors. Hence, the vertex u has at least 3 non-neighbors[3] (since there are at least 5 vertices distinct from u in total). Thus, we can find three distinct non-neighbors p, q and r of u. Consider these p, q and r. If all of pq, qr and rp are edges of G, then G has a triangle (namely, $\{p, q, r\}$). If not, then G has an anti-triangle (for example, if pq is not an edge of G, then $\{u, p, q\}$ is an anti-triangle). In either case, we are thus done with the proof in Case 2. Thus, both cases are resolved, and the proof is complete. $\square$

Notice the symmetry between Case 1 and Case 2 in our above proof: the arguments used were almost the same, except that neighbors and non-neighbors swapped roles.

Remark 2.3.4. Proposition 2.3.1 could also be proved by brute force as well (using a computer). Indeed, it clearly suffices to prove it for all simple graphs with 6 vertices (as opposed to ≥ 6 vertices), because if a graph has more than 6 vertices, then we can just throw away some of them until we have only 6 left. However, there are only finitely many simple graphs with 6 vertices (up to relabeling of their vertices), and the validity of Proposition 2.3.1 can be checked for each of them. This is, of course, cumbersome (even a computer would take a moment checking all the 2^{15} possible graphs for triangles and anti-triangles) and unenlightening.

Proposition 2.3.1 is the first result in a field of graph theory known as **Ramsey theory**. I shall not dwell on this field in this course, but let me make a few more remarks. The first step beyond Proposition 2.3.1 is the following generalization:

Proposition 2.3.5. Let r and s be two positive integers. Let G be a simple graph with $|\operatorname{V}(G)| \geq \dbinom{r+s-2}{r-1}$. Then, at least one of the following two statements holds:

[3] The word "non-neighbor" shall here mean a vertex that is not adjacent to u and **distinct from** u. Thus, u does not count as a non-neighbor of u.

- *Statement 1:* There exist r distinct vertices of G that are mutually adjacent (i.e., each two distinct ones among these r vertices are adjacent).
- *Statement 2:* There exist s distinct vertices of G that are mutually non-adjacent (i.e., no two distinct ones among these s vertices are adjacent).

Applying Proposition 2.3.5 to $r = 3$ and $s = 3$, we can recover Proposition 2.3.1.

One might wonder whether the number $\dbinom{r+s-2}{r-1}$ in Proposition 2.3.5 can be improved – i.e., whether we can replace it by a smaller number without making Proposition 2.3.5 false. In the case of $r = 3$ and $s = 3$, this is impossible, because the number 6 in Proposition 2.3.1 cannot be made smaller[4]. However, for some other values of r and s, the value $\dbinom{r+s-2}{r-1}$ can be improved. (For example, for $r = 4$ and $s = 4$, the best possible value is 18 rather than $\dbinom{4+4-2}{4-1} = 20$.) The smallest possible value that could stand in place of $\dbinom{r+s-2}{r-1}$ in Proposition 2.3.5 is called the **Ramsey number** $R(r,s)$; thus, we have just showed that $R(3,3) = 6$. Finding $R(r,s)$ for higher values of r and s is a hard computational challenge; here are some values that have been found with the help of computers:

$$R(3,4) = 9; \qquad R(3,5) = 14; \qquad R(3,6) = 18; \qquad R(3,7) = 23;$$
$$R(3,8) = 28; \qquad R(3,9) = 36; \qquad R(4,4) = 18; \qquad R(4,5) = 25.$$

(We are only considering the cases $r \leq s$, since it is easy to see that $R(r,s) = R(s,r)$ for all r and s. Also, the trivial values $R(1,s) = 1$ and $R(2,s) = s+1$ for $s \geq 2$ are omitted.) The Ramsey number $R(5,5)$ is still unknown (although it is known that $43 \leq R(5,5) \leq 48$).

Proposition 2.3.5 can be further generalized to a result called *Ramsey's theorem*. The idea behind the generalization is to slightly change the point of view, and replace the simple graph G by a complete graph (i.e., a simple graph in which every two distinct vertices are adjacent) whose edges are colored in two colors (say, blue and red). This is a completely equivalent concept, because the concepts of "adjacent" and "non-adjacent" in G can be identified with the concepts of "adjacent through a blue edge" (i.e., the edge connecting them is colored blue) and "adjacent through a red edge", respectively. Statements 1 and 2 then turn into "there exist r distinct vertices that are mutually adjacent through blue edges" and "there exist s distinct vertices that are mutually adjacent through red edges", respectively. From this point of view, it is only logical

[4]Indeed, we saw in Example 2.3.3 **(c)** that 5 vertices would not suffice.

to generalize Proposition 2.3.5 further to the case when the edges of a complete graph are colored in k (rather than two) colors. The corresponding generalization is known as Ramsey's theorem. We refer to the well-written Wikipedia page `https://en.wikipedia.org/wiki/Ramsey's_theorem` for a treatment of this generalization with proof, as well as a table of known Ramsey numbers $R(r, s)$ and a self-contained (if somewhat terse) proof of Proposition 2.3.5. Ramsey's theorem can be generalized and varied further; this usually goes under the name "Ramsey theory". For elementary introductions, see the Cut-the-knot page `http://www.cut-the-knot.org/Curriculum/Combinatorics/ThreeOrThree.shtml`, the above-mentioned Wikipedia article, as well as the texts by Harju [Harju14], Bollobas [Bollob98] and West [West01].

There is one more direction in which Proposition 2.3.1 can be improved a bit: A graph G with at least 6 vertices has not only one triangle or anti-triangle, but at least two of them (this can include having one triangle and one anti-triangle). Proving this makes for a nice exercise:

Exercise 2.1. Let G be a simple graph. A **triangle-or-anti-triangle** in G means a set that is either a triangle or an anti-triangle.

(a) Assume that $|\operatorname{V}(G)| \geq 6$. Prove that G has at least two triangle-or-anti-triangles. (For comparison: Proposition 2.3.1 shows that G has at least one triangle-or-anti-triangle.)

(b) Assume that $|\operatorname{V}(G)| = m + 6$ for some $m \in \mathbb{N}$. Prove that G has at least $m + 1$ triangle-or-anti-triangles.

[**Solution:** This is Exercise 1 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

2.4. Degrees

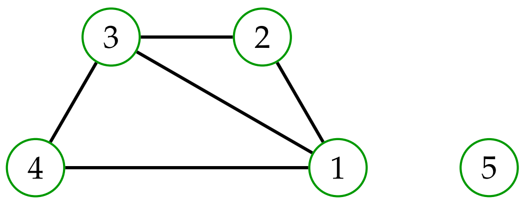
The **degree** of a vertex in a simple graph just counts how many edges contain this vertex:

Definition 2.4.1. 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

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

(These equalities are pretty easy to check: Each edge $e \in E$ that contains v contains exactly one neighbor of v, and conversely, each neighbor of v belongs to exactly one edge that contains v. However, these equalities are specific to simple graphs, and won't hold any more once we move on to multigraphs.)

For example, in the graph



,

the vertices have degrees

$$\deg 1 = 3, \qquad \deg 2 = 2, \qquad \deg 3 = 3, \qquad \deg 4 = 2, \qquad \deg 5 = 0.$$

Here are some basic properties of degrees in simple graphs:

Proposition 2.4.2. 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 $(n-1)$-element set $\operatorname{V}(G) \setminus \{v\}$. Thus, their number is $\leq n-1$. $\square$

Proposition 2.4.3 (Euler 1736)**.** Let G be a simple graph. 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. Write the simple graph G as $G = (V, E)$; thus, $\operatorname{V}(G) = V$ and $\operatorname{E}(G) = E$.

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

1. We can obtain N by computing, 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 $\sum\limits_{v \in V} \deg v$.

2. On the other hand, we can obtain N by computing, for each $e \in E$, the number of all $v \in V$ that satisfy $v \in e$, and summing these numbers over all e. Since each $e \in E$ contains exactly 2 vertices $v \in V$, this result will be $\sum\limits_{e \in E} 2 = |E| \cdot 2 = 2 \cdot |E|$.

Since these two results must be equal (because they both equal N), we thus see that $\sum\limits_{v \in V} \deg v = 2 \cdot |E|$. But this is the claim of Proposition 2.4.3. $\square$

Corollary 2.4.4 (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. Proposition 2.4.3 yields that $\sum\limits_{v \in \mathrm{V}(G)} \deg v = 2 \cdot |\mathrm{E}(G)|$. Hence, $\sum\limits_{v \in \mathrm{V}(G)} \deg v$ is even. However, if a sum of integers is even, then it must have an even number of odd addends. Thus, the sum $\sum\limits_{v \in \mathrm{V}(G)} \deg v$ must have an even number of odd addends. In other words, the number of vertices v of G whose degree $\deg v$ is odd is even. $\square$

Corollary 2.4.4 is often stated as follows: In a group of people, the number of persons with an odd number of friends (in the group) is even. It is also known as the **handshake lemma**.

Here is another property of degrees in a simple graph:

Proposition 2.4.5. 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. So the degrees of all n vertices of G are distinct, where $n = |\mathrm{V}(G)|$.

In other words, the map

$$\begin{aligned} \deg : \mathrm{V}(G) &\to \{0, 1, \ldots, n-1\}, \\ v &\mapsto \deg v \end{aligned}$$

is injective. But this is a map between two finite sets of the same size (n). When such a map is injective, it has to be bijective (by the pigeonhole principle). Therefore, in particular, it takes both 0 and $n-1$ as values.

In other words, there are a vertex u with degree 0 and a vertex v with degree $n-1$. Are these two vertices adjacent or not? Yes because of $\deg v = n-1$; no because of $\deg u = 0$. Contradiction!

(Fine print: The two vertices u and v must be distinct, since $0 \neq n-1$. It is here that we are using the "at least two vertices" assumption!) $\square$

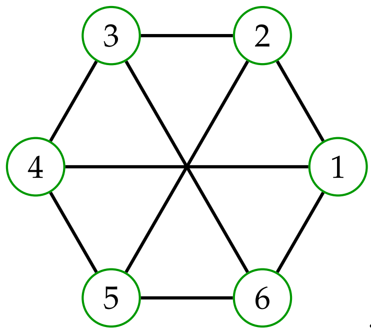
Here is an application of counting neighbors to proving a fact about graphs. This is known as **Mantel's theorem**:

Theorem 2.4.6 (Mantel's theorem)**.** Let G be a simple graph with n vertices and e edges. Assume that $e > n^2/4$. Then, G has a triangle (i.e., three distinct vertices that are pairwise adjacent).

Example 2.4.7. Let G be the graph (V, E), where

$$V = \{1, 2, 3, 4, 5, 6\} ;$$
$$E = \{12,\ 23,\ 34,\ 45,\ 56,\ 61,\ 14,\ 25,\ 36\} .$$

Here is a drawing:



.

This graph has no triangle (which, by the way, is easy to verify without checking all possibilities: just observe that every edge of G joins two vertices of different parity, but a triangle would necessarily have two vertices of equal parity). Thus, by the contrapositive of Mantel's theorem, it satisfies $e \leq n^2/4$ with $n = 6$ and $e = 9$. This is indeed true because $9 = 6^2/4$. But this also entails that if we add any further edge to G, then we obtain a triangle.

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. Let $V = \operatorname{V}(G)$ and $E = \operatorname{E}(G)$, so that $G = (V, E)$.

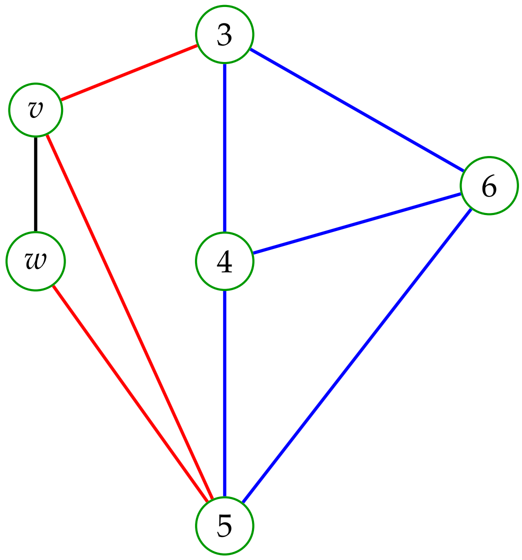
We must prove that G has a triangle. Assume the contrary. Thus, G has no triangle.

From $e > n^2/4 \geq 0$, we see that G has an edge. Pick any such edge, and call it vw. Thus, $v \neq w$.

Let us now color each edge of G with one of three colors, as follows:

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

The following picture shows an example of this coloring:



We now count the edges of each color:

- There is exactly 1 black edge – namely, vw.
- How many red edges can there be? I claim that there are at most $n-2$. Indeed, each vertex other than v and w is connected to at most one of v and w by a red edge, since otherwise it would form a triangle with v and w.
- How many blue edges can there be? The vertices other than v and w, along with the blue edges that join them, form a graph with $n-2$ vertices; this graph has no triangles (since G has no triangles). By the induction hypothesis, however, if this graph had more than $(n-2)^2/4$ edges, then it would have a triangle. Thus, it has $\leq (n-2)^2/4$ edges. In other words, there are $\leq (n-2)^2/4$ blue edges.

In total, the number of edges is therefore

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

In other words, $e \leq n^2/4$. This contradicts $e > n^2/4$. This is the contradiction we were looking for, so the induction is complete. $\square$

Quick question: What about equality? Can a graph with n vertices and exactly $n^2/4$ edges have no triangles? Yes (for even n). Indeed, for any even n, we can take the graph

$$(\{1,2,\ldots,n\}, \{ij \mid i \not\equiv j \bmod 2\})$$

(keep in mind that ij means the 2-element set $\{i, j\}$ here, not the product $i \cdot j$). We can also do this for odd n, and obtain a graph with $\left(n^2 - 1\right)/4$ edges (which is as close to $n^2/4$ as we can get when n is odd – after all, the number of edges has to be an integer). So the bound in Mantel's theorem is optimal (as far as integers are concerned).

The following exercise can be regarded as a "mirror version" of Mantel's theorem:

Exercise 2.2. Let G be a simple graph with n vertices and e edges. Assume that $e < n(n-2)/4$. Prove that G has an anti-triangle (i.e., three distinct vertices that are pairwise non-adjacent).

[**Solution:** This is Exercise 2 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

Mantel's theorem can be generalized:

Theorem 2.4.8 (Turan's theorem)**.** Let r be a positive integer. Let G be a simple graph with n vertices and e edges. Assume that

$$e > \frac{r-1}{r} \cdot \frac{n^2}{2}.$$

Then, there exist $r+1$ distinct vertices of G that are mutually adjacent.

Mantel's theorem is the particular case for $r = 2$. We will see a proof of Turan's theorem later (Theorem 7.3.1). Mantel's and Turan's theorems are two of the simplest results of **extremal graph theory** – the study of how inequalities between some graph parameters (in our case: the numbers of vertices and edges) imply the existence of certain substructures (in our case: of a triangle or of $r+1$ mutually adjacent vertices). Deeper introductions to this subject can be found in [Zhao23, Chapters 1 and 5] and [Jukna11].

Exercise 2.3. Let $G = (V, E)$ be a simple graph. Set $n = |V|$. Prove that we can find some edges $e_1, e_2, \ldots, e_k$ of G and some triangles $t_1, t_2, \ldots, t_\ell$ of G such that $k + \ell \leq n^2/4$ and such that each edge $e \in E \setminus \{e_1, e_2, \ldots, e_k\}$ is a subset of (at least) one of the triangles $t_1, t_2, \ldots, t_\ell$.

[**Remark:** In other words, this exercise is claiming that all edges of G can be covered by at most $n^2/4$ edge-or-triangles. Here, an **edge-or-triangle** means either an edge or a triangle of G, and the word "covers" means that each edge of G is a subset of the chosen edge-or-triangles.]

[**Hint:** Imitate the above proof of Mantel's theorem.]

Remark 2.4.9. Exercise 2.3 is a generalization of Mantel's theorem. Indeed, if the simple graph $G = (V, E)$ has no triangles, then the number ℓ in Exercise 2.3 must be 0, and thus the edges $e_1, e_2, \ldots, e_k$ must be all edges of G, so that we conclude that $|E| = k \leq k + \ell \leq n^2/4$.

Exercise 2.4. Let G be a simple graph with n vertices and k edges, where $n > 0$. Prove that G has at least $\dfrac{k}{3n}\left(4k - n^2\right)$ triangles.

[**Hint:** First argue that for any edge vw of G, the total number of triangles that contain v and w is at least $\deg v + \deg w - n$. Then, use the inequality $n\left(a_1^2 + a_2^2 + \cdots + a_n^2\right) \geq \left(a_1 + a_2 + \cdots + a_n\right)^2$, which holds for any n real numbers $a_1, a_2, \ldots, a_n$. (This is a particular case of the Cauchy–Schwarz inequality or the Chebyshev inequality or the Jensen inequality – pick your favorite!)]

Remark 2.4.10. Exercise 2.4 is known as the **Moon–Moser inequality for triangles**. It, too, generalizes Mantel's theorem: If $k > n^2/4$, then $\dfrac{k}{3n}\left(4k - n^2\right) > 0$, and therefore Exercise 2.4 entails that G has at least one triangle.

Exercise 2.5. Let $G = (V, E)$ be a simple graph.

An edge $e = \{u, v\}$ of G will be called **odd** if the number $\deg u + \deg v$ is odd.

Prove that the number of odd edges of G is even.

[**Hint:** There are several solutions. One uses modular arithmetic and (in particular) the congruence $m^2 \equiv m \mod 2$ for every integer m. Other solutions use nothing but common sense.]

Exercise 2.6. Let $G = (V, E)$ be a simple graph. Let S be a subset of V, and let $k = |S|$. Prove that

$$\sum_{v \in S} \deg v \leq k(k-1) + \sum_{v \in V \setminus S} \min\{\deg v, k\}.$$

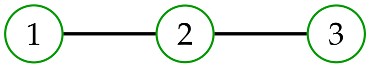
Remark 2.4.11. Exercise 2.6 has a converse (the so-called **Erdös–Gallai theorem**): If $d_1, d_2, \ldots, d_n$ are n nonnegative integers such that $d_1 + d_2 + \cdots + d_n$ is even and such that $d_1 \geq d_2 \geq \cdots \geq d_n$ and such that each $k \in \{1, 2, \ldots, n\}$ satisfies

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{d_i, k\},$$

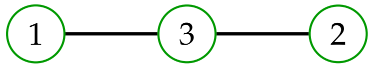
then there exists a simple graph with vertex set $\{1, 2, \ldots, n\}$ whose vertices have degrees $d_1, d_2, \ldots, d_n$.

2.5. Graph isomorphism

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



and



.

Let us formalize this:

Definition 2.5.1. Let G and H be two simple graphs.

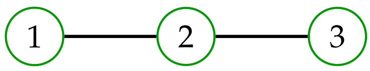
(a) A **graph isomorphism** (or **isomorphism**) from G to H means a bijection $\phi : \operatorname{V}(G) \to \operatorname{V}(H)$ that "preserves edges", i.e., that has the following property: For any two vertices u and v of G, we have

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

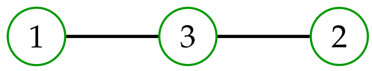
(b) We say that G and H are **isomorphic** (this is written $G \cong H$) if there exists a graph isomorphism from G to H.

Here are two examples:

- The two graphs

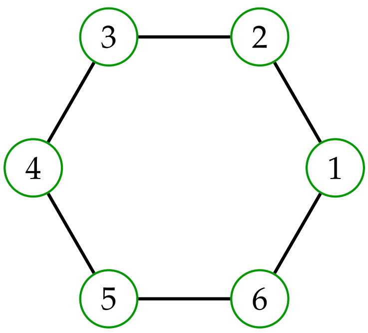


and

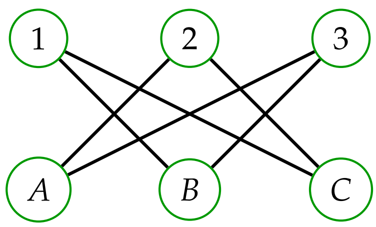


are isomorphic, because the bijection between their vertex sets that sends $1,2,3$ to $1,3,2$ is an isomorphism. Another isomorphism between the same two graphs sends $1,2,3$ to $2,3,1$.

- The two graphs



and



are isomorphic, because the bijection between their vertex sets that sends $1,2,3,4,5,6$ to $1,B,3,A,2,C$ is an isomorphism.

Here are some basic properties of isomorphisms (the proofs are straightforward):

Proposition 2.5.2. Let G and H be two graphs. The inverse of a graph isomorphism ϕ from G to H is a graph isomorphism from H to G.

Proposition 2.5.3. Let G, H and I be three graphs. If ϕ is a graph isomorphism from G to H, and ψ is a graph isomorphism from H to I, then $\psi \circ \phi$ is a graph isomorphism from G to I.

As a consequence of these two propositions, it is easy to see that the relation $\cong$ (on the class of all graphs) is an equivalence relation.

Graph isomorphisms preserve all "intrinsic" properties of a graph. For example:

Proposition 2.5.4. Let G and H be two simple graphs, and ϕ a graph isomorphism from G to H. Then:

- **(a)** For every $v \in \operatorname{V}(G)$, we have $\deg_G v = \deg_H (\phi(v))$. Here, $\deg_G v$ means the degree of v as a vertex of G, whereas $\deg_H (\phi(v))$ means the degree of $\phi(v)$ as a vertex of H.
- **(b)** We have $|\operatorname{E}(H)| = |\operatorname{E}(G)|$.
- **(c)** We have $|\operatorname{V}(H)| = |\operatorname{V}(G)|$.

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

Proposition 2.5.5. 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. □

2.6. Some families of graphs

We will now define some particularly significant families of graphs.

2.6.1. Complete and empty graphs

The simplest families of graphs are the complete graphs and the empty graphs:

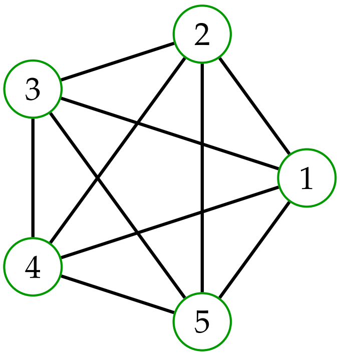
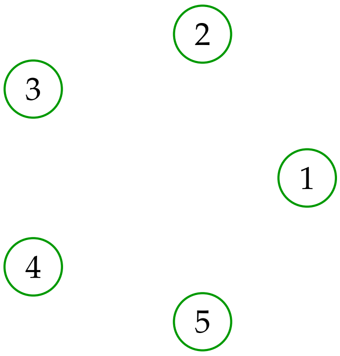
Definition 2.6.1. Let V be a finite set.

(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 the complete graph on V is denoted K_n.

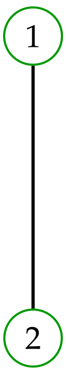
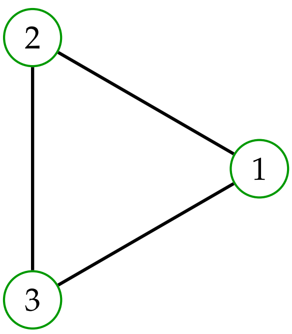
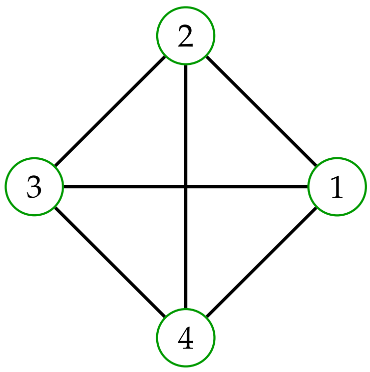
(b) The **empty graph** on V means the simple graph $(V, \varnothing)$. It is the simple graph with vertex set V and no edges.

The following pictures show the complete graph and the empty graph on the set $\{1, 2, 3, 4, 5\}$:

complete graph	empty graph
	

The complete one is called K_5.

Here are the complete graphs K_0, K_1, K_2, K_3, K_4:

K_0	K_1	K_2	K_3	K_4
				

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: If $|V| \neq |W|$, then there are none. If $|V| = |W|$, then any bijection from V to W is an isomorphism. The same holds for empty graphs.

2.6.2. Path and cycle graphs

Next come two families of graphs with fairly simple shapes:

Definition 2.6.2. For each $n \in \mathbb{N}$, we define the ***n*-th path graph** P_n to be the simple graph

$$\begin{aligned}&(\{1,2,\ldots,n\},\ \ \{\{i,i+1\}\ \mid\ 1 \leq i < n\}) \\ &= (\{1,2,\ldots,n\},\ \ \{12,\ 23,\ 34,\ \ldots,\ (n-1)\,n\}).\end{aligned}$$

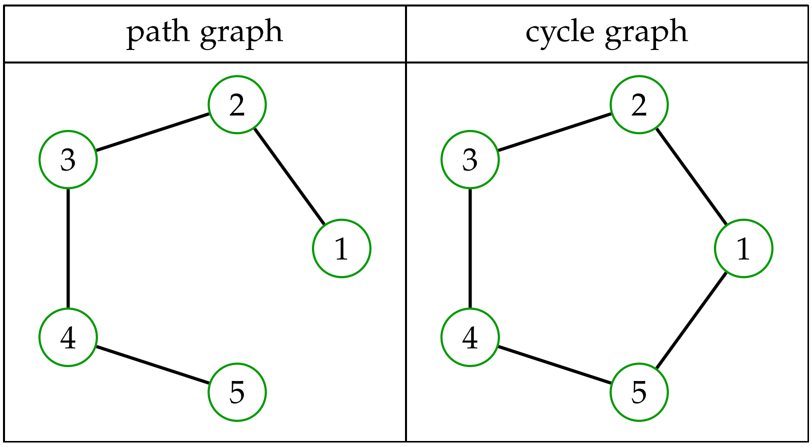
This graph has n vertices and $n-1$ edges (unless $n = 0$, in which case it has 0 edges).

Definition 2.6.3. For each $n > 1$, we define the ***n*-th cycle graph** C_n to be the simple graph

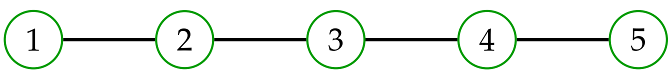
$$\begin{aligned}&(\{1,2,\ldots,n\},\ \ \{\{i,i+1\}\ \mid\ 1 \leq i < n\} \cup \{\{n,1\}\}) \\ &= (\{1,2,\ldots,n\},\ \ \{12,\ 23,\ 34,\ \ldots,\ (n-1)\,n,\ n1\}).\end{aligned}$$

This graph has n vertices and n edges (unless $n = 2$, in which case it has 1 edge only). (We will later modify the definition of the 2-nd cycle graph C_2 somewhat, in order to force it to have 2 edges. But we cannot do this yet, since a simple graph with 2 vertices cannot have 2 edges.)

The following pictures show the path graph P_5 and the cycle graph C_5:



Of course, it is more common to draw the path graph stretched out horizontally:



Note that the cycle graph C_3 is identical with the complete graph K_3.

Question: What are the graph isomorphisms from P_n to itself?
Answer: One such isomorphism is the identity map $\operatorname{id} : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$. Another is the "reversal" map

$$\begin{aligned} \{1, 2, \ldots, n\} &\to \{1, 2, \ldots, n\}, \\ i &\mapsto n + 1 - i. \end{aligned}$$

There are no others.

Question: What are the graph isomorphisms from C_n to itself?
Answer: For any $k \in \mathbb{Z}$, we can define a "rotation by k vertices", which is the map

$$\begin{aligned} &\{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}, \\ &\qquad i \mapsto (i + k \text{ reduced modulo } n \text{ to an element of } \{1, 2, \ldots, n\}). \end{aligned}$$

Thus we get n rotations (one for each $k \in \{1, 2, \ldots, n\}$); all of them are graph isomorphisms.

There are also the reflections, which are the maps

$$\begin{aligned} &\{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}, \\ &\qquad i \mapsto (k - i \text{ reduced modulo } n \text{ to an element of } \{1, 2, \ldots, n\}) \end{aligned}$$

for $k \in \mathbb{Z}$. There are n of them, too, and they are isomorphisms as well.

Altogether we obtain $2n$ isomorphisms (for $n > 2$), and there are no others. (The group they form is the n-th dihedral group.)

2.6.3. Kneser graphs

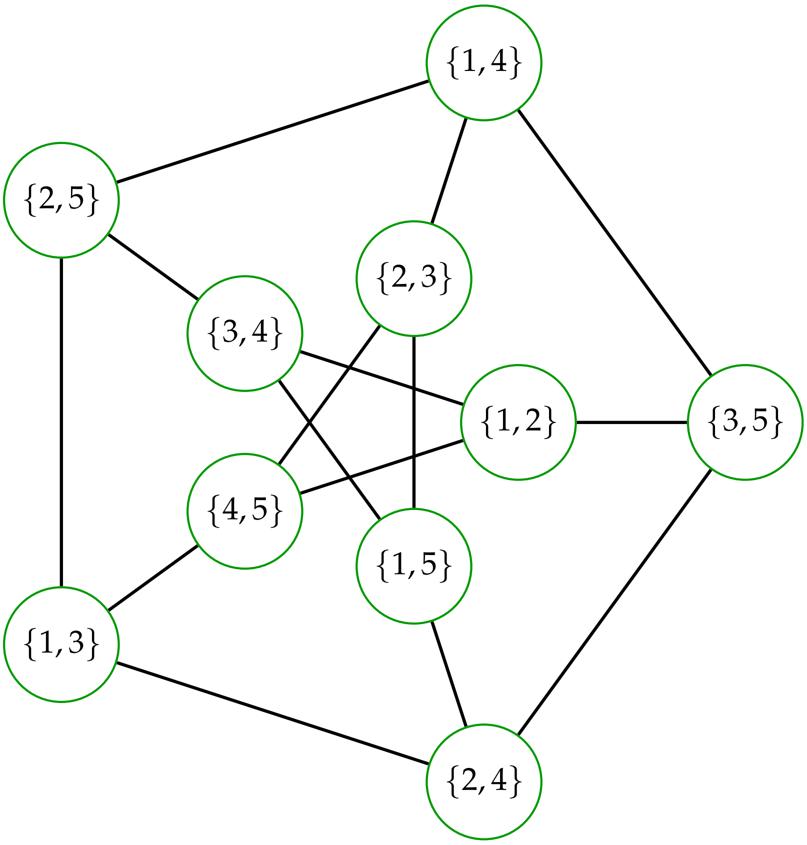
Here is a more exotic family of graphs:

Example 2.6.4. 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

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

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

The graph $K_{\{1,2,\ldots,5\},2}$ is called the **Petersen graph**; here is how it looks like:



2.7. Subgraphs

Definition 2.7.1. 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

$$(S, \quad E \cap \mathcal{P}_2(S))$$

of G. In other words, it denotes 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, of course, that no edges remain "dangling" – i.e., if you throw away a vertex, then you must throw away all edges that contain this vertex). Such a subgraph is an induced subgraph if no edges are removed without need – i.e., if you removed only those edges that lost some of their endpoints. Thus, induced subgraphs can be characterized as follows:

Proposition 2.7.2. 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 belong to $\operatorname{V}(H)$ is an edge of H.

Proof. This is a matter of understanding the definition. □

Example 2.7.3. Let $n > 1$ be an integer.

(a) The path graph P_n is a subgraph of the cycle graph C_n. It is not an induced subgraph (for $n > 2$), because it contains the two vertices n and 1 of C_n but does not contain the edge $n1$.

(b) The path graph P_{n-1} is an induced subgraph of P_n. (Namely, it is the induced subgraph of P_n on the set $\{1, 2, \ldots, n-1\}$.)

(c) Assume that $n > 3$. Is C_{n-1} a subgraph of C_n ? No, because the edge $(n-1)\,1$ belongs to C_{n-1} but not to C_n.

The following is easy:

Proposition 2.7.4. Let G be a simple graph, and let H be a subgraph of G. Assume that H is a complete graph. Then, H is automatically an induced subgraph of G.

Proof. This follows from Proposition 2.7.2, since the completeness of H means that each 2-element subset $\{u, v\}$ of the vertex set of H is an edge of H. □

We note that triangles in a graph can be characterized in terms of complete subgraphs. Namely, a triangle "is" the same as a complete subgraph (or, equivalently, induced complete subgraph) with three vertices:

Remark 2.7.5. Let G be a simple graph. Let u, v, w be three distinct vertices of G. The following are equivalent:

1. The set $\{u, v, w\}$ is a triangle of G.
2. The induced subgraph of G on $\{u, v, w\}$ is isomorphic to K_3.
3. The induced subgraph of G on $\{u, v, w\}$ is isomorphic to C_3.

Thus, instead of saying "triangle of G", one often says "a K_3 in G" or "a C_3 in G". Generally, "an H in G" (where H and G are two graphs) means a subgraph of G that is isomorphic to H. (In the case when $H = K_3 = C_3$, it does not matter whether we require it to be a subgraph or an induced subgraph, since a complete subgraph has to be induced automatically.)

Exercise 2.7. Let n be a positive integer. Let S be a simple graph with $2n$ vertices. Prove that S has two distinct vertices that have an even number of common neighbors.

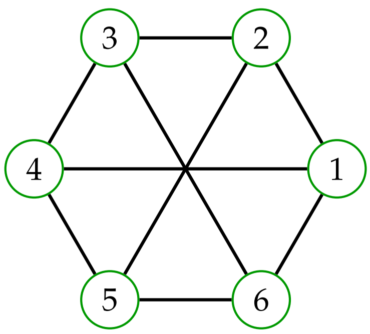
Exercise 2.8. Let $n \geq 2$ be an integer. Let G be a simple graph with n vertices.

(a) Describe G if the degrees of the vertices of G are $1, 1, \ldots, 1, n-1$.

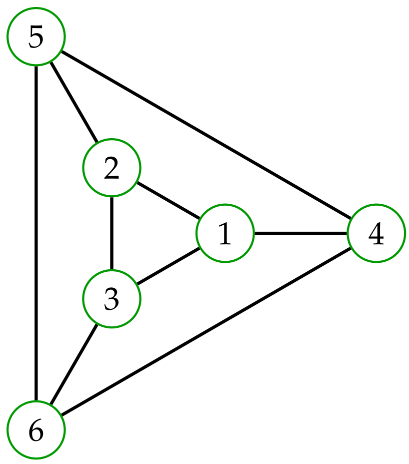
(b) Let a and b be two positive integers such that $a + b = n$. Describe G if the degrees of the vertices of G are $1, 1, \ldots, 1, a, b$.

Here, to "describe" G means to explicitly determine (with proof) a graph that is isomorphic to G.

Remark 2.7.6. The situations in Exercise 2.8 are, in a sense, exceptional. Typically, the degrees of the vertices of a graph do not uniquely determine the graph up to isomorphism. For example, the two graphs



and

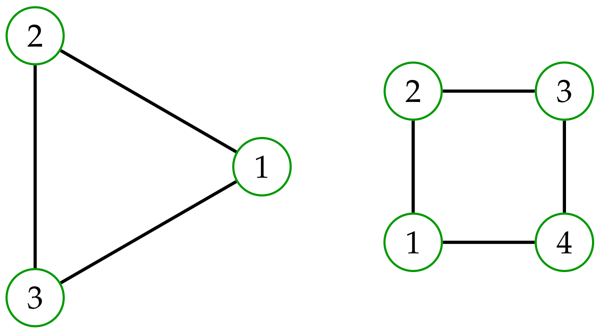


are not isomorphic[5], but have the same degrees (namely, each vertex of either graph has degree 3).

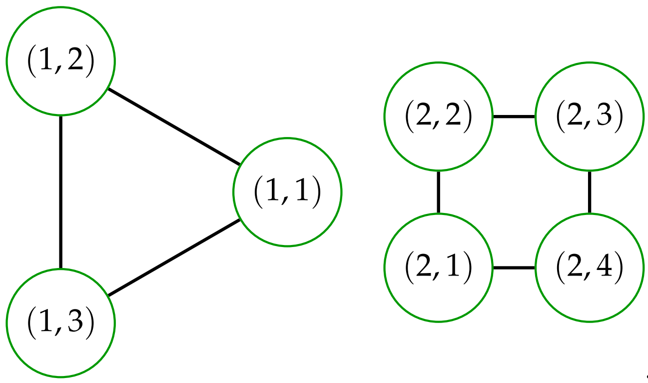
[5] The easiest way to see this is to observe that the second graph has a triangle (i.e., three distinct vertices that are mutually adjacent), while the first graph does not.

2.8. Disjoint unions

Another 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 simple 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 this formally watertight, we have to relabel each vertex v of each graph G_i as the pair (i, v), so that vertices coming from different graphs appear as different even if they were equal. For example, the disjoint union $C_3 \sqcup C_4$ of the two cycle graphs C_3 and C_4 should not be



(which makes no sense, because there are two points labelled 1 in this picture, but a graph can have only one vertex 1), but rather should be



.

So here is the formal definition:

Definition 2.8.1. Let $G_1, G_2, \ldots, G_k$ be simple graphs, where $G_i = (V_i, E_i)$ for each $i \in \{1, 2, \ldots, k\}$. The **disjoint union** of these k graphs $G_1, G_2, \ldots, G_k$ is defined to be the simple graph (V, E), where

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

This disjoint union is denoted by $G_1 \sqcup G_2 \sqcup \cdots \sqcup G_k$.

Note: If G and H are two graphs, then the two graphs $G \sqcup H$ and $H \sqcup G$ are isomorphic, but not the same graph (unless $G = H$). For example, $C_3 \sqcup C_4$ has a vertex $(2, 4)$, but $C_4 \sqcup C_3$ does not.

2.9. Walks and paths

We now come to the definitions of walks and paths – two of the most fundamental features that graphs can have. In particular, Euler's 1736 paper, where graphs were first studied, is about certain kinds of walks.

2.9.1. 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 often get from a town to another even if they are not adjacent. This is made formal in the concept of a "walk":

Definition 2.9.1. 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 defined to be $v_0, v_1, \ldots, v_k$.
- The **edges** of $\mathbf{w}$ are defined to be $v_0v_1,\ v_1v_2,\ v_2v_3,\ \ldots,\ v_{k-1}v_k$.
- The nonnegative integer k is called the **length** of $\mathbf{w}$. (This is the number of all edges of $\mathbf{w}$, counted with multiplicity. It is 1 smaller than the number of all vertices of $\mathbf{w}$, counted with multiplicity.)
- The vertex v_0 is called the **starting point** of $\mathbf{w}$. We say that **w starts** (or **begins**) at v_0.
- The vertex v_k is called the **ending point** of $\mathbf{w}$. We say that **w ends** at v_k.

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

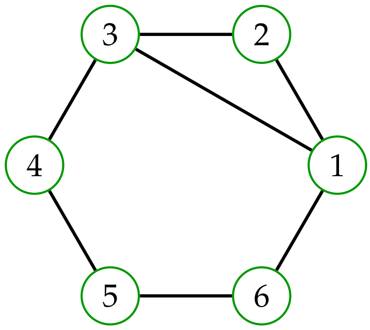
(d) Let p and q be two vertices of G. A **walk from p to q** means a walk that starts at p and ends at q. A **path from p to q** means a path that starts at p and ends at q.

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

Example 2.9.2. Let G be the graph

$$(\{1,2,3,4,5,6\},\ \{12,\ 23,\ 34,\ 45,\ 56,\ 61,\ 13\}).$$

This graph looks as follows:



Then:

- The sequence $(1, 3, 4, 5, 6, 1, 3, 2)$ of vertices of G is a walk in G. This walk is a walk from 1 to 2. It is not a path. The length of this walk is 7.
- The sequence $(1, 2, 4, 3)$ of vertices of G is not a walk, since 24 is not an edge of G. Hence, it is not a path either.
- The sequence $(1, 3, 2, 1)$ is a walk from 1 to 1. It has length 3. It is not a path.
- The sequence $(1, 2, 1)$ is a walk from 1 to 1. It has length 2. It is not a path.
- The sequence (5) is a walk from 5 to 5. It has length 0. It is a path. More generally, each vertex v of G produces a length-0 path (v).
- The sequence $(5, 4)$ is a walk from 5 to 4. It has length 1. It is a path. More generally, each edge uv of G produces a length-1 path (u, v).

Intuitively, we can think of walks and paths as follows:

- A **walk** of a graph is a way of walking from one vertex to another (or to the same vertex) by following a sequence of edges.
- A **path** is a walk whose vertices are distinct (i.e., each vertex appears at most once in the walk).

Exercise 2.9. Let G be a simple graph. Let $\mathbf{w}$ be a path in G. Prove that the edges of $\mathbf{w}$ are distinct. (This may look obvious when you can point to a picture; but we ask you to give a rigorous proof!)

[**Solution:** This is Exercise 3 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

2.9.2. Composing/concatenating and reversing walks

Here are some simple things we can do with walks and paths.

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

Proposition 2.9.3. Let G be a simple graph. Let u, v and w be three vertices of G. 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,

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

is a walk from u to w. This walk shall be denoted $\mathbf{a} * \mathbf{b}$.

Proof. Intuitively clear and straightforward to verify. □

Proposition 2.9.4. 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. Then:

(a) The list $(a_k, a_{k-1}, \ldots, a_0)$ is a walk from v to u. We denote this walk by $\operatorname{rev} \mathbf{a}$ and call it the **reversal** of $\mathbf{a}$.

(b) If $\mathbf{a}$ is a path, then $\operatorname{rev} \mathbf{a}$ is a path again.

Proof. Intuitively clear and straightforward to verify. □

2.9.3. Reducing walks to paths

A path is just a walk without repeated vertices. If you have a walk, you can turn it into a path by removing "loops" (or "digressions"):

Proposition 2.9.5. 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 $a_i = a_j$. Consider these i and j. Now, consider the tuple

$$\left(\underbrace{a_0, a_1, \ldots, a_i}_{\text{the first } i+1 \text{ vertices of } \mathbf{a}}, \underbrace{a_{j+1}, a_{j+2}, \ldots, a_k}_{\text{the last } k-j \text{ vertices of } \mathbf{a}} \right)$$

(this is just $\mathbf{a}$ with the part between a_i and a_j cut out). This tuple is a walk from u to v, and its length is $\underbrace{i}_{<j} + (k-j) < j + (k-j) = k$. So we have found a walk from u to v whose length is smaller than k. This proves the proposition. □

Example 2.9.6. Consider the walk $(1,3,4,5,6,1,3,2)$ from Example 2.9.2. Then, Proposition 2.9.5 tells us that there is a walk from 1 to 2 that has smaller length. You can find this walk by removing the part between the two 3's. You get the walk $(1,3,2)$. This is actually a path.

Corollary 2.9.7 (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. Proposition 2.9.5 says that if there is a walk from u to v that is not a path, then there is a walk from u to v having shorter length. Apply this repeatedly, until you get a path. (You will eventually get a path, because the length cannot keep decreasing forever.) $\square$

2.9.4. Remark on algorithms

We take a little break from proving structural theorems in order to address some important computational questions. As always in these notes, we will only scratch the surface and content ourselves with simple but not quite optimal algorithms.

Given a simple graph G and two vertices u and v of G, we can ask ourselves the following questions:

> **Question 1:** Does G have a walk from u to v ?
>
> **Question 2:** Does G have a path from u to v ?
>
> **Question 3:** Find a shortest path from u to v (that is, a path from u to v having the smallest possible length), or determine that no such path exists.
>
> **Question 4:** Given a number $k \in \mathbb{N}$, find a walk from u to v having length k, or determine that no such walk exists.
>
> **Question 5:** Given a number $k \in \mathbb{N}$, find a path from u to v having length k, or determine that no such path exists.

Corollary 2.9.7 reveals that Questions 1 and 2 are equivalent (indeed, the existence of a walk from u to v entails the existence of a path from u to v by Corollary 2.9.7, whereas the converse is obvious). Question 3 is clearly a stronger version of Question 2 (in the sense that any answer to Question 3 will automatically answer Question 2 as well).

With a bit more thought, it is easily seen that Question 4 is a stronger version of Question 3. Indeed, Corollary 2.9.7 shows that a shortest walk from u to v (if it exists) must also be a shortest path from u to v. However, any path from u to v must have length $\leq n-1$, where n is the number of vertices of G (since a path of length k has $k + 1$ distinct vertices, but G has only n vertices to spare). Hence, if there is no walk of length $\leq n - 1$ from u to v, then there is no path from u to v whatsoever. Thus, if we answer Question 4 for all values $k \in \{0, 1, \ldots, n-1\}$, then we obtain either a shortest path from u to v (by taking the smallest k for which the answer is positive, and then picking the resulting walk, which must be a shortest path by what we previously said), or proof positive that no path from u to v exists (if the answer for each $k \in \{0, 1, \ldots, n-1\}$ is negative).

Thus, answering Question 4 will yield answers to Questions 1, 2 and 3.

Let us now outline a way how Question 4 can be answered using a recursive algorithm. Specifically, we recurse on k. The base case ($k = 0$) is easy: A walk from u to v having length 0 exists if $u = v$ and does not exist otherwise. The interesting part is the recursion step: Assume that the integer k is positive, and that we already know how to answer Question 4 for $k - 1$ instead of k. Now, let us answer it for k. To do so, we observe that any walk from u to v having length k must have the form $(u, \ldots, w, v)$, where the penultimate vertex w is some neighbor of v. Moreover, if we remove the last vertex v from our walk $(u, \ldots, w, v)$, then we obtain a walk $(u, \ldots, w)$ of length $k - 1$. Hence, we can find a walk from u to v having length k as follows:

- We make a list of all neighbors of v. We go through this list in some arbitrary order.
- For each neighbor w in this list, we try to find a walk from u to w having length $k - 1$ (this is a matter of answering Question 4 for $k - 1$ instead of k, so we supposedly already know how to do this). If such a walk exists, then we simply insert v at its end, and thus obtain a walk from u to v having length k. Thus we obtain a positive answer to our question.
- If we have gone through our whole list of neighbors of v without finding a walk from u to v having length k, then no such walk exists, and thus we have found a negative answer.

This recursive algorithm answers Question 4, and is fast enough to be practically viable if implemented well. (In the language of complexity theory, it is a polynomial time algorithm[6].) Much more efficient algorithms exist, however. In applications, a generalized version of Question 3 often appears, asking for a path that is shortest not in the sense of smallest length, but in the sense of smallest "weighted length" (i.e., different edges contribute differently to this "length"). This generalized question is one of the most fundamental algorithmic problems in computer science, known as the **shortest path problem**, and various algorithms can be found on its Wikipedia page and in algorithm-focussed texts such as [Griffi21, §3.5], [KelTro17, §12.3], [Schrij17, Chapter 1] or (for a royal treatment) [Schrij03, Chapters 6–8].

[6]To be specific: Its running time can be bounded in a polynomial of n and k, where n is the number of vertices of G.

Question 5 looks superficially similar to Question 4, yet it differs in the most important way: There is no efficient algorithm known for answering it! In the language of complexity theory, it is an NP-hard problem, which means that a polynomial-time algorithm for it is not expected to exist (although this is the kind of negative that appears near-impossible to prove at the current stage of the discipline). It is still technically a finite problem (there are only finitely many possible paths in G, and thus one can theoretically try them all), and there is even a polynomial-time algorithm for any fixed value of k (again, a trivial one: check all the n^{k+1} possible $(k+1)$-tuples of vertices of G for whether they are paths from u to v), but the complexity of this algorithm grows exponentially in k, which makes it useless in practice.

2.9.5. The equivalence relation "path-connected"

We can use the concepts of walks and paths to define a certain equivalence relation on the vertex set $\operatorname{V}(G)$ of any graph G:

Definition 2.9.8. 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 of G, we shall have $u \simeq_G v$ if and only if there exists a walk 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 2.9.9. Let G be a simple graph. Then, the relation $\simeq_G$ is an equivalence relation.

Proof. We need to show that $\simeq_G$ is symmetric, reflexive and transitive.

- **Symmetry:** If $u \simeq_G v$, then $v \simeq_G u$, because we can take a walk from u to v and reverse it.
- **Reflexivity:** We always have $u \simeq_G u$, since the trivial walk (u) is a walk from u to u.
- **Transitivity:** If $u \simeq_G v$ and $v \simeq_G w$, then $u \simeq_G w$, because (as we know from Proposition 2.9.3) we can take a walk $\mathbf{a}$ from u to v and a walk $\mathbf{b}$ from v to w and combine them to form the walk $\mathbf{a} * \mathbf{b}$ defined in Proposition 2.9.3.

$\square$

Proposition 2.9.10. Let G be a simple graph. Let u and v be two vertices of G. Then, $u \simeq_G v$ if and only if there exists a path from u to v.

Proof. $\Longleftarrow$: Clear, since any path is a walk.

$\Longrightarrow$: This is just saying that if there is a walk from u to v, then there is a path from u to v. But this follows from Corollary 2.9.7. $\square$

2.9.6. Connected components and connectedness

The equivalence relation $\simeq_G$ introduced in Definition 2.9.8 allows us to define two important concepts:

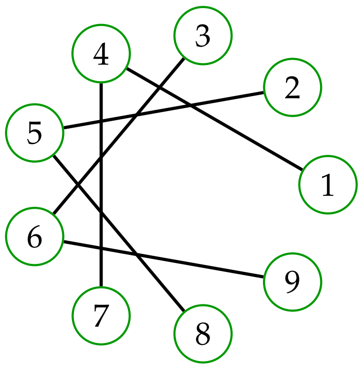
Definition 2.9.11. Let G be a simple graph. The equivalence classes of the equivalence relation $\simeq_G$ are called the **connected components** (or, for short, **components**) of G.

Definition 2.9.12. Let G be a simple graph. We say that G is **connected** if G has exactly one component.

Thus, a simple graph G is connected if and only if it has at least one component (i.e., it has at least one vertex) and it has at most one component (i.e., each two of its vertices are path-connected).

Example 2.9.13. Let G be the graph with vertex set $\{1, 2, \ldots, 9\}$ and such that two vertices i and j are adjacent if and only if $|i - j| = 3$. What are the components of G ?

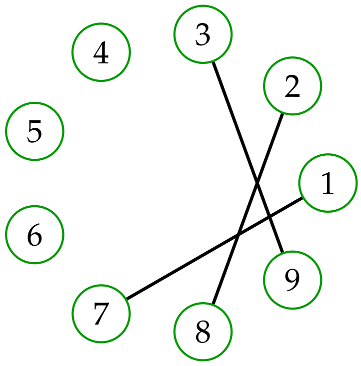
The graph G looks like this:



.

This looks like a jumbled mess, so you might think that all vertices are mutually path-connected. But this is not the case, because edges that cross in a drawing do not necessarily have endpoints in common. Walks can only move from one edge to another at a common endpoint. Thus, there are much fewer walks than the picture might suggest. We have $1 \simeq_G 4 \simeq_G 7$ and $2 \simeq_G 5 \simeq_G 8$ and $3 \simeq_G 6 \simeq_G 9$, but there are no further $\simeq_G$-relations. In fact, two vertices of G are adjacent only if they are congruent modulo 3 (as numbers), and therefore you cannot move from one modulo-3 congruence class to another by walking along edges of G. So the components of G are $\{1, 4, 7\}$ and $\{2, 5, 8\}$ and $\{3, 6, 9\}$. The graph G is not connected.

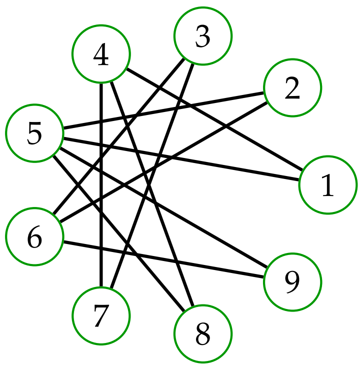
Example 2.9.14. Let G be the graph with vertex set $\{1, 2, \ldots, 9\}$ and such that two vertices i and j are adjacent if and only if $|i - j| = 6$. This graph looks like this:



.

What are the components of G ? They are $\{1,7\}$ and $\{2,8\}$ and $\{3,9\}$ and $\{4\}$ and $\{5\}$ and $\{6\}$. Note that three of these six components are singleton sets. The graph G is not connected.

Example 2.9.15. Let G be the graph with vertex set $\{1,2,\ldots,9\}$ and such that two vertices i and j are adjacent if and only if $|i-j|=3$ or $|i-j|=4$. This graph looks like this:



.

We can take a long walk through G:

$$(1,4,7,3,6,9,5,2,5,8).$$

This walk traverses every vertex of G; thus, any two vertices of G are path-connected. Hence, G has only one component, namely $\{1,2,\ldots,9\}$. Thus, G is connected.

Example 2.9.16. The complete graph on a nonempty set is connected. The complete graph on the empty set is not connected, since it has 0 (not 1) components.

Example 2.9.17. The empty graph on a finite set V has $|V|$ many components (those are the singleton sets $\{v\}$ for $v\in V$). Thus, it is connected if and only if $|V|=1$.

Exercise 2.10. Let $k \in \mathbb{N}$. Let S be a finite set.

Recall that the **Kneser graph** $K_{S,k}$ is the simple graph whose vertices are the k-element subsets of S, and whose edges are the unordered pairs $\{A, B\}$ consisting of two such subsets A and B that satisfy $A \cap B = \varnothing$.

Prove that this Kneser graph $K_{S,k}$ is connected if $|S| \geq 2k + 1$.

[**Remark:** Can the "if" here be replaced by an "if and only if"? Not quite, because the graph $K_{S,k}$ is also connected if $|S| = 2$ and $k = 1$ (in which case it has two vertices and one edge), or if $|S| = k$ (in which case it has only one vertex), or if $k = 0$ (in which case it has only one vertex). But these are the only "exceptions".]

2.9.7. Induced subgraphs on components

The following is not hard to see:

Proposition 2.9.18. Let G be a simple graph. Let C be a component of G.

Then, the induced subgraph of G on the set C is connected.

Proof. Let $G[C]$ be this induced subgraph. We need to show that $G[C]$ is connected. In other words, we need to show that $G[C]$ has exactly 1 component.

Clearly, $G[C]$ has at least one vertex (since C is a component, i.e., an equivalence class of $\simeq_G$, but equivalence classes are always nonempty), thus has at least 1 component. So we only need to show that $G[C]$ has no more than 1 component. In other words, we need to show that any two vertices of $G[C]$ are path-connected in $G[C]$.

So let u and v be two vertices of $G[C]$. Then, $u, v \in C$, and therefore $u \simeq_G v$ (since C is a component of G). In other words, there exists a walk $\mathbf{w} = (w_0, w_1, \ldots, w_k)$ from u to v in G. We shall now prove that this walk $\mathbf{w}$ is actually a walk of $G[C]$. In other words, we shall prove that all vertices of $\mathbf{w}$ belong to C.

But this is easy: If w_i is a vertex of $\mathbf{w}$, then $(w_0, w_1, \ldots, w_i)$ is a walk from u to w_i in G, and therefore we have $u \simeq_G w_i$, so that w_i belongs to the same component of G as u; but that component is C. Thus, we have shown that each vertex w_i of $\mathbf{w}$ belongs to C. Therefore, $\mathbf{w}$ is a walk of the graph $G[C]$. Consequently, it shows that $u \simeq_{G[C]} v$.

We have now proved that $u \simeq_{G[C]} v$ for any two vertices u and v of $G[C]$. Hence, the relation $\simeq_{G[C]}$ has no more than 1 equivalence class. In other words, the graph $G[C]$ has no more than 1 component. This completes our proof. $\square$

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 2.9.19. Let G be a simple graph. Let $C_1, C_2, \ldots, C_k$ be all components of G (listed without repetition).

Thus, G is isomorphic to the disjoint union $G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k]$.

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) of $G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k]$ to the vertex v of G. We claim that this bijection is a graph isomorphism. In order to prove this, we need to check that there are no edges of G that join vertices in different components. But this is easy: If two vertices in different components of G were adjacent, then they would be path-connected, and thus would actually belong to the same component. $\square$

The upshot of these results is that every simple graph can be decomposed into a disjoint union of its components (or, more precisely, of the induced subgraphs on its components). Each of these components is a connected graph. Moreover, this is easily seen to be the only way to decompose the graph into a disjoint union of connected graphs.

2.9.8. Some exercises on connectedness

Exercise 2.11. Let G be a simple graph with $\operatorname{V}(G) \neq \varnothing$. Show that the following two statements are equivalent:

- *Statement 1:* The graph G is connected.
- *Statement 2:* For every two nonempty subsets A and B of $\operatorname{V}(G)$ satisfying $A \cap B = \varnothing$ and $A \cup B = \operatorname{V}(G)$, there exist $a \in A$ and $b \in B$ such that $ab \in \operatorname{E}(G)$. (In other words: Whenever we subdivide the vertex set $\operatorname{V}(G)$ of G into two nonempty subsets, there will be at least one edge of G connecting a vertex in one subset to a vertex in another.)

[**Solution:** This is Exercise 7 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

Exercise 2.12. Let V be a nonempty finite set. Let G and H be two simple graphs such that $\operatorname{V}(G) = \operatorname{V}(H) = V$. Assume that for each $u \in V$ and $v \in V$, there exists a path from u to v in G or a path from u to v in H. Prove that at least one of the graphs G and H is connected.

[**Solution:** This is Exercise 8 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

Exercise 2.13. Let $G = (V, E)$ be a simple graph. The **complement graph** $\overline{G}$ of G is defined to be the simple graph $(V, \mathcal{P}_2(V) \setminus E)$. (Thus, two distinct vertices u and v in V are adjacent in $\overline{G}$ if and only if they are not adjacent in G.)

Prove that at least one of the following two statements holds:

- *Statement 1:* For each $u \in V$ and $v \in V$, there exists a path from u to v in G of length ≤ 3.
- *Statement 2:* For each $u \in V$ and $v \in V$, there exists a path from u to v in $\overline{G}$ of length ≤ 2.

[**Solution:** This is Exercise 9 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

Exercise 2.14. Let $n \geq 2$ be an integer. Let G be a connected simple graph with n vertices.

(a) Describe G if the degrees of the vertices of G are $1, 1, 2, 2, \ldots, 2$ (exactly two 1's and $n - 2$ many 2's).

(b) Describe G if the degrees of the vertices of G are $1, 1, \ldots, 1, n - 1$.

(c) Describe G if the degrees of the vertices of G are $2, 2, \ldots, 2$.

Here, to "describe" G means to explicitly determine (with proof) a graph that is isomorphic to G.

The following exercise is not explicitly concerned with connectedness and components, but it might help to think about components to solve it (although there are solutions that do not use them):

Exercise 2.15. Let G be a simple graph with n vertices. Assume that each vertex of G has at least one neighbor.

A **matching** of G shall mean a set F of edges of G such that no two edges in F have a vertex in common. Let m be the largest size of a matching of G.

An **edge cover** of G shall mean a set F of edges of G such that each vertex of G is contained in at least one edge $e \in F$. Let c be the smallest size of an edge cover of G.

Prove that $c + m = n$.

Remark 2.9.20. Let G be the cycle graph C_5 shown in Example 2.13.2. Then, $\{12, 34\}$ is a matching of G of largest possible size (why?), whereas $\{12, 34, 25\}$ is an edge cover of G of smallest possible size (why?). Thus, Exercise 2.15 says that $2 + 3 = 5$ here, which is indeed true.

2.10. Closed walks and cycles

Here are two further kinds of walks:

Definition 2.10.1. Let G be a simple graph.

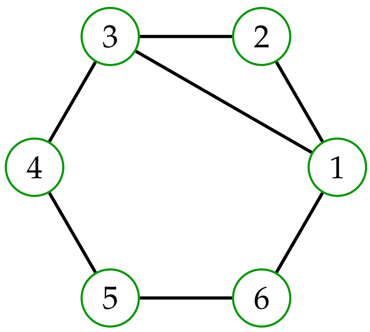
(a) A **closed walk** of G means a walk whose first vertex is identical with its last vertex. In other words, it means a walk $(w_0, w_1, \ldots, w_k)$ with $w_0 = w_k$. Sometimes, closed walks are also known as **circuits** (but many authors use this latter word for something slightly different).

(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.

Example 2.10.2. Let G be the simple graph

$$(\{1, 2, 3, 4, 5, 6\}, \ \{12, \ 23, \ 34, \ 45, \ 56, \ 61, \ 13\}).$$

This graph looks as follows (we have already seen it in Example 2.9.2):



Then:

- The sequence $(1, 3, 2, 1, 6, 5, 6, 1)$ is a closed walk of G. But it is very much not a cycle.
- The sequences $(1, 2, 3, 1)$ and $(1, 3, 4, 5, 6, 1)$ and $(1, 2, 3, 4, 5, 6, 1)$ are cycles of G. You can get further cycles by rotating these sequences (in a proper sense of this word – e.g., rotating $(1, 2, 3, 1)$ gives $(2, 3, 1, 2)$ and $(3, 1, 2, 3)$) and by reversing them. Every cycle of G can be obtained in this way.
- The sequences (1) and $(1, 2, 1)$ are closed walks, but not cycles of G (since they fail the $k \geq 3$ condition).
- The sequence $(1, 2, 3)$ is a walk, but not a closed walk, since $1 \neq 3$.

Authors have different opinions about whether $(1,2,3,1)$ and $(1,3,2,1)$ count as different cycles. Fortunately, this matters only if you want to count cycles, but not for the existence or non-existence of cycles.

We have now defined paths (in an arbitrary graph) and also path graphs P_n; we have also defined cycles (in an arbitrary graph) and also cycle graphs C_n. Besides their similar names, are they related? The answer is "yes":

Proposition 2.10.3. 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 the path graph P_{k+1}, namely the subgraph $(\{p_0, p_1, \ldots, p_k\}, \{p_i p_{i+1} \mid 0 \leq i < k\})$. (If this subgraph is actually an induced subgraph of G, then the path $(p_0, p_1, \ldots, p_k)$ is called an "induced path".)

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 the cycle graph C_k, namely the subgraph $(\{c_0, c_1, \ldots, c_k\}, \{c_i c_{i+1} \mid 0 \leq i < k\})$. (If this subgraph is actually an induced subgraph of G, then the cycle $(c_0, c_1, \ldots, c_k)$ is called an "induced cycle".)

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

Proof. Straightforward. □

Certain graphs contain cycles; other graphs don't. For instance, the complete graph K_n contains a lot of cycles (when $n \geq 3$), whereas the path graph P_n contains none. Let us try to find some criteria for when a graph can and when it cannot have cycles[7]:

Proposition 2.10.4. Let G be a simple graph. Let $\mathbf{w}$ be a walk of G such that no two adjacent edges of $\mathbf{w}$ are identical. (By "adjacent edges", we mean edges of the form $w_{i-1}w_i$ and $w_i w_{i+1}$, where w_{i-1}, w_i, w_{i+1} are three consecutive vertices of $\mathbf{w}$.)

Then, $\mathbf{w}$ either is a path or contains a cycle (i.e., there exists a cycle of G whose edges are edges of $\mathbf{w}$).

Example 2.10.5. Let G be as in Example 2.10.2. Then, $(2,1,3,2,1,6)$ is a walk $\mathbf{w}$ of G such that no two adjacent edges of $\mathbf{w}$ are identical (even though the edge 21 appears twice in this walk). On the other hand, $(2,1,3,1,6)$ is not such a walk (since its two adjacent edges 13 and 31 are identical).

[7] Mantel's theorem already gives such a criterion for cycles of length 3 (because a cycle of length 3 is the same as a triangle).

Proof of Proposition 2.10.4. We assume that $\mathbf{w}$ is not a path. We must then show that $\mathbf{w}$ contains a cycle.

Write $\mathbf{w}$ as $\mathbf{w} = (w_0, w_1, \ldots, w_k)$. Since $\mathbf{w}$ is not a path, two of the vertices $w_0, w_1, \ldots, w_k$ must be equal. In other words, there exists a pair (i, j) of integers i and j with $i < j$ and $w_i = w_j$. Among all such pairs, we pick one with **minimum** difference $j - i$. We shall show that the walk $(w_i, w_{i+1}, \ldots, w_j)$ is a cycle.

First, this walk is clearly a closed walk (since $w_i = w_j$). It thus remains to show that $j - i \geq 3$ and that the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct. The distinctness of $w_i, w_{i+1}, \ldots, w_{j-1}$ follows from the minimality of $j - i$. To show that $j - i \geq 3$, we assume the contrary. Thus, $j - i$ is either 1 or 2 (since $i < j$). But $j - i$ cannot be 1, since the endpoints of an edge cannot be equal (since our graph is a simple graph). So $j - i$ must be 2. Thus, $w_i = w_{i+2}$. Therefore, the two edges $w_i w_{i+1}$ and $w_{i+1} w_{i+2}$ are identical. But this contradicts the fact that no two adjacent edges of $\mathbf{w}$ are identical. Contradiction, qed. □

Corollary 2.10.6. Let G be a simple graph. Assume that G has a closed walk $\mathbf{w}$ of length > 0 such that no two adjacent edges of $\mathbf{w}$ are identical. Then, G has a cycle.

Proof. This follows from Proposition 2.10.4, since $\mathbf{w}$ is not a path. □

Theorem 2.10.7. Let G be a simple graph. Let u and v be two vertices in G. Assume that there are two distinct paths from u to v. Then, G has a cycle.

Proof. More generally, we shall prove this theorem with the word "path" replaced by "backtrack-free walk", where a "**backtrack-free walk**" means a walk $\mathbf{w}$ such that no two adjacent edges of $\mathbf{w}$ are identical. This is a generalization of the theorem, since every path is a backtrack-free walk (why?).

So we claim the following:

> *Claim 1:* Let $\mathbf{p}$ and $\mathbf{q}$ be two distinct backtrack-free walks that start at the same vertex and end at the same vertex. Then, G has a cycle.

We shall prove Claim 1 by induction on the length of $\mathbf{p}$. So we fix an integer N, and we assume that Claim 1 is proved in the case when the length of $\mathbf{p}$ is $N - 1$. We must now show that it is also true when the length of $\mathbf{p}$ is N.

So let $\mathbf{p} = (p_0, p_1, \ldots, p_a)$ and $\mathbf{q} = (q_0, q_1, \ldots, q_b)$ be two distinct backtrack-free walks that start at the same vertex and end at the same vertex and satisfy $a = N$. We must find a cycle.

The walks $\mathbf{p}$ and $\mathbf{q}$ are distinct but start at the same vertex, so they cannot both be trivial[8]. If one of them is trivial, then the other is a closed walk (because a trivial walk is a closed walk), and then our goal follows from Corollary 2.10.6

[8] We say that a walk is **trivial** if it has length 0.

in this case (because we have a nontrivial closed backtrack-free walk). Hence, from now on, we WLOG assume that neither of the two walks **p** and **q** is trivial. Thus, each of these two walks has a last edge. The last edge of **p** is $p_{a-1}p_a$, whereas the last edge of **q** is $q_{b-1}q_b$.

Two cases are possible:

Case 1: We have $p_{a-1}p_a = q_{b-1}q_b$.

Case 2: We have $p_{a-1}p_a \neq q_{b-1}q_b$.

Let us consider Case 1 first. In this case, the last edges $p_{a-1}p_a$ and $q_{b-1}q_b$ of the two walks **p** and **q** are identical, so the second-to-last vertices of these two walks must also be identical. Thus, if we remove these last edges from both walks, then we obtain two shorter backtrack-free walks $(p_0, p_1, \ldots, p_{a-1})$ and $(q_0, q_1, \ldots, q_{b-1})$ that again start at the same vertex and end at the same vertex, but the length of the first of them is $a - 1 = N - 1$. Hence, by the induction hypothesis, we can apply Claim 1 to these two shorter walks (instead of **p** and **q**), and we conclude that G has a cycle. So we are done in Case 1.

Let us now consider Case 2. In this case, we combine the two walks **p** and **q** (more precisely, **p** and the reversal of **q**) to obtain the closed walk

$$(p_0, p_1, \ldots, p_{a-1}, p_a = q_b, q_{b-1}, \ldots, q_0).$$

This closed walk is backtrack-free (since $(p_0, p_1, \ldots, p_a)$ and $(q_0, q_1, \ldots, q_b)$ are backtrack-free, and since $p_{a-1}p_a \neq q_{b-1}q_b$) and has length > 0 (since it contains at least the edge $p_{a-1}p_a$). Hence, Corollary 2.10.6 entails that G has a cycle.

We have thus found a cycle in both Cases 1 and 2. This completes the induction step. Thus, we have proved Claim 1. As we said, Theorem 2.10.7 follows from it. ☐

Exercise 2.16. Let G be a simple graph.

- **(a)** Prove that if G has a closed walk of odd length, then G has a cycle of odd length.
- **(b)** Is it true that if G has a closed walk of length not divisible by 3, then G has a cycle of length not divisible by 3 ?
- **(c)** Does the answer to part **(b)** change if we replace "walk" by "non-backtracking walk"? (A walk **w** with edges $e_1, e_2, \ldots, e_k$ (in this order) is said to be **non-backtracking** if each $i \in \{1, 2, \ldots, k-1\}$ satisfies $e_i \neq e_{i+1}$.)
- **(d)** A **trail** (in a graph) means a walk whose edges are distinct (but whose vertices are not necessarily distinct). Does the answer to part **(b)** change if we replace "walk" by "trail"?

(Proofs and counterexamples should be given.)

2.11. The longest path trick

Here is another proposition that guarantees the existence of cycles in a graph under certain circumstances. More importantly, its proof illustrates a useful tactic in dealing with graphs:

> **Proposition 2.11.1.** 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 G have a longest path? Let's see: Any path of G has length $\leq |V| - 1$, since its vertices have to be distinct. Moreover, G has at least one vertex and thus has at least one path. A finite nonempty set of integers has a largest element. Thus, G has a longest path.)

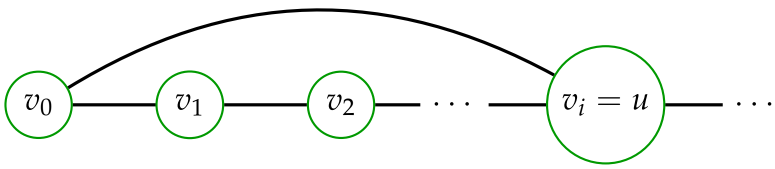
The vertex v_0 has degree $\geq d$ (by assumption), and thus has $\geq d$ neighbors (since the degree of a vertex is the number of its neighbors).

If all neighbors of v_0 belonged to the set $\{v_1, v_2, \ldots, v_{d-1}\}$ [9], then the number of neighbors of v_0 would be at most $d-1$, which would contradict the previous sentence. Thus, there exists at least one neighbor u of v_0 that does **not** belong to this set $\{v_1, v_2, \ldots, v_{d-1}\}$. Consider this u. Then, $u \neq v_0$ (since a vertex cannot be its own neighbor).

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

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

If we had $u \notin \{v_0, v_1, \ldots, v_m\}$, then this walk $\mathbf{p}'$ would be a path; but this would contradict the fact that $\mathbf{p}$ is a **longest** path of G. Thus, we must have $u \in \{v_0, v_1, \ldots, v_m\}$. In other words, $u = v_i$ for some $i \in \{0, 1, \ldots, m\}$. Consider this i. Since $u \neq v_0$ and $u \notin \{v_1, v_2, \ldots, v_{d-1}\}$, we thus have $i \geq d$. Here is a picture:



Now, consider the walk

$$\mathbf{c} := (u, v_0, v_1, \ldots, v_i).$$

This is a closed walk (since $u = v_i$) and has length $i + 1 \geq d + 1$ (since $i \geq d$). If we can show that $\mathbf{c}$ is a cycle, then we have thus found a cycle of length $\geq d+1$, so we will be done.

[9] If $d - 1 > m$, then this set should be understood to mean $\{v_1, v_2, \ldots, v_m\}$.

It thus remains to prove that $\mathbf{c}$ is a cycle. Let us do this. We need to check that the vertices $u, v_0, v_1, \ldots, v_{i-1}$ are distinct, and that the length of $\mathbf{c}$ is ≥ 3. The latter claim is clear: The length of $\mathbf{c}$ is $i + 1 \geq d + 1 \geq 3$ (since $d > 1$ and $d \in \mathbb{Z}$). The former claim is not much harder: Since $u = v_i$, the vertices $u, v_0, v_1, \ldots, v_{i-1}$ are just the vertices $v_i, v_0, v_1, \ldots, v_{i-1}$, and thus are distinct because they are distinct vertices of the path $\mathbf{p}$. The proof of Proposition 2.11.1 is thus complete. □

2.12. Bridges

One question that will later prove crucial is: What happens to a graph if we remove a single edge from it? Let us first define a notation for this:

Definition 2.12.1. 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 this edge e. In other words,

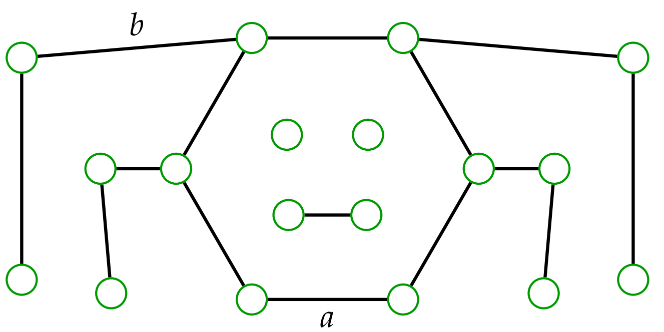
$$G \setminus e := (V, \ E \setminus \{e\}) .$$

Some authors write $G - e$ for $G \setminus e$.

Theorem 2.12.2. 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 of vertices. It is these sets that we are talking about here, not the induced subgraphs on these sets.)
- **(b)** If e appears in no cycle of G (in other words, there exists no cycle of G such that e is an edge of this cycle), then the graph $G \setminus e$ has one more component than G.

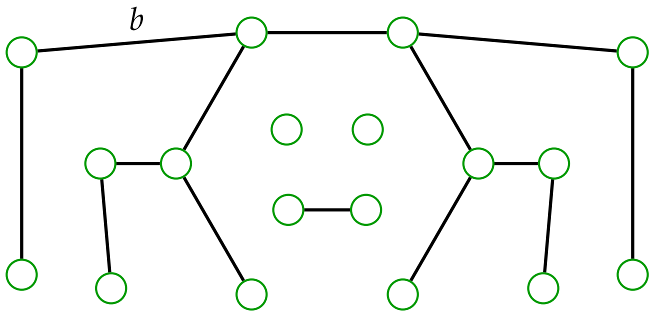
Example 2.12.3. Let G be the graph shown in the following picture:



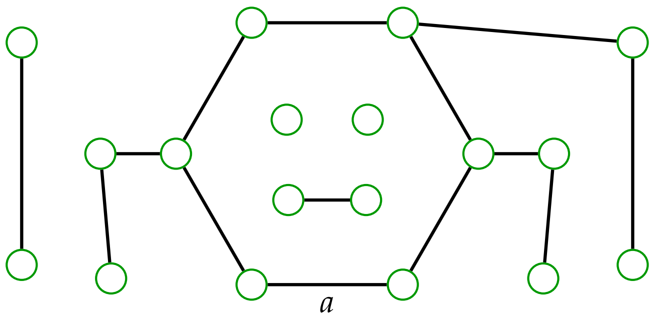
(1)

(where we have labeled the edges a and b for further reference). This graph has 4 components. The edge a is an edge of a cycle of G, whereas the edge

b appears in no cycle of G. Thus, if we set $e = a$, then Theorem 2.12.2 **(a)** shows that the components of $G \setminus e$ are precisely the components of G. This graph $G \setminus e$ for $e = a$ looks as follows:



and visibly has 4 components. On the other hand, if we set $e = b$, then Theorem 2.12.2 **(b)** shows that the graph $G \setminus e$ has one more component than G. This graph $G \setminus e$ for $e = b$ looks as follows:



and visibly has 5 components.

Proof of Theorem 2.12.2. We will only sketch the proof. For details, see [21f6, §6.7].

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

(a) Assume that e is an edge of some cycle of G. Then, if you remove e from this cycle, then you still have a path from u to v left (as the remaining edges of the cycle function as a detour), and this path is a path of $G \setminus e$. Thus, $u \simeq_{G \setminus e} v$.

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 relation $\simeq_{G \setminus e}$ is precisely the relation $\simeq_G$ (because the components of a graph are the equivalence classes of its $\simeq$ relation). So let us prove the latter fact.

We must show that two vertices x and y of G satisfy $x \simeq_{G \setminus e} y$ if and only if they satisfy $x \simeq_G y$. The "only if" part is obvious (since a walk of $G \setminus e$ is always a walk of G). It thus remains to prove the "if" part. So we assume that x and y are two vertices of G satisfying $x \simeq_G y$, and we want to show that $x \simeq_{G \setminus e} y$.

From $x \simeq_G y$, we conclude that G has a path from x to y (by Proposition 2.9.10). If this path does not use[10] the edge e, then it is a path from x to y in $G \setminus e$, and thus we have $x \simeq_{G \setminus e} y$, which is what we wanted to prove. So we WLOG assume that this path does use the edge e. Thus, this path contains the endpoints u and v of this edge e. We WLOG assume that u appears before v on this path (otherwise, just swap u with v). Thus, this path looks as follows:

$$(x, \ldots, u, v, \ldots, y).$$

If we remove the edge $e = uv$, then this path breaks into two smaller paths

$$(x, \ldots, u) \qquad \text{and} \qquad (v, \ldots, y)$$

(since the edges of a path are distinct, so e appears only once in it). Both of these two smaller paths are paths of $G \setminus e$. Thus, $x \simeq_{G \setminus e} u$ and $v \simeq_{G \setminus e} y$. Now, recalling that $\simeq_{G \setminus e}$ is an equivalence relation, we combine these results to obtain

$$x \simeq_{G \setminus e} u \simeq_{G \setminus e} v \simeq_{G \setminus e} y.$$

Hence, $x \simeq_{G \setminus e} y$. This completes the proof of Theorem 2.12.2 **(a)**.

(b) Assume that e appears in no cycle of G. We must prove that the graph $G \setminus e$ 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 component exists, 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 e). Thus, they remain components of $G \setminus e$. (Formalizing this is a nice exercise in formalization; see [21f6, §6.7].)

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.

Then, we must show that $A \cup B = C$ and $A \cap B = \varnothing$.

To see that $A \cap B = \varnothing$, we need to show that $u \simeq_{G \setminus e} v$ does **not** hold (since A and B are the equivalence classes of u and v with respect to the relation $\simeq_{G \setminus e}$). So let us do this. Assume the contrary. Thus, $u \simeq_{G \setminus e} v$. Hence, there exists a

[10] We say that a walk **w** **uses** an edge f if f is an edge of **w**.

path from u to v in $G \setminus e$. Since $e = uv$, we can "close" this path by appending the vertex u to its end; the result is a cycle of the graph G that contains the edge e. But this contradicts our assumption that no cycle of G contains e. This contradiction shows that our assumption was wrong. Thus, we conclude that $u \simeq_{G \setminus e} v$ does **not** hold. Hence, as we said, $A \cap B = \varnothing$.

It remains to show that $A \cup B = C$. Since A and B are clearly subsets of C (because each walk of $G \setminus e$ is a walk of G, and thus each component of $G \setminus e$ is a subset of a component of G), we have $A \cup B \subseteq C$, and therefore we only need to show that $C \subseteq A \cup B$. In other words, we need to show that each $c \in C$ belongs to $A \cup B$.

Let us show this. Let $c \in C$ be a vertex. Then, $c \simeq_G u$ (since C is the component of G containing u). Therefore, G has a path $\mathbf{p}$ from c to u. Consider this path $\mathbf{p}$. Two cases are possible:

- *Case 1:* This path $\mathbf{p}$ does not use the edge e. In this case, $\mathbf{p}$ is a path of $G \setminus e$, and thus we obtain $c \simeq_{G \setminus e} u$. In other words, $c \in A$ (since A is the component of $G \setminus e$ containing u).
- *Case 2:* This path $\mathbf{p}$ does use the edge e. In this case, the edge e must be the last edge of $\mathbf{p}$ (since the path $\mathbf{p}$ would otherwise contain the vertex u twice[11]; but a path cannot contain a vertex twice), and the last two vertices of $\mathbf{p}$ must be v and u in this order. Thus, by removing the last vertex from $\mathbf{p}$, we obtain a path from c to v, and this latter path is a path of $G \setminus e$ (since it no longer contains u and therefore does not use e). This yields $c \simeq_{G \setminus e} v$. In other words, $c \in B$ (since B is the component of $G \setminus e$ containing v).

In either of these two cases, we have shown that c belongs to one of A and B. In other words, $c \in A \cup B$. This is precisely what we wanted to show. This completes the proof of Theorem 2.12.2 **(b)**. $\square$

We introduce some fairly standard terminology:

Definition 2.12.4. Let e be an edge of a simple graph G.

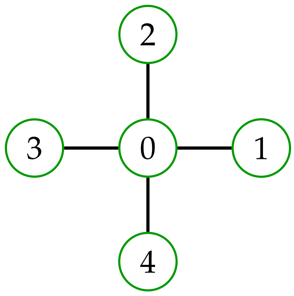
- **(a)** We say that e is a **bridge** (of G) if e appears in no cycle of G.
- **(b)** We say that e is a **cut-edge** (of G) if the graph $G \setminus e$ has more components than G.

Corollary 2.12.5. Let e be an edge of a simple graph G. Then, e is a bridge if and only if e is a cut-edge.

Proof. Follows from Theorem 2.12.2. $\square$

[11] Indeed, the path $\mathbf{p}$ already ends in u. If it would contain e anywhere other than at the very end, then it would thus contain the vertex u twice (since u is an endpoint of e).

We can also define "cut-vertices": A vertex v of a graph G is said to be a **cut-vertex** if the graph $G \setminus v$ (that is, the graph G with the vertex v removed[12]) has more components than G. Unfortunately, there doesn't seem to be an analogue of Corollary 2.12.5 for cut-vertices. Note also that removing a vertex (unlike removing an edge) can add more than one component to the graph (or it can also subtract 1 component if this vertex had degree 0). For example, removing the vertex 0 from the graph



results in an empty graph on the set $\{1,2,3,4\}$, so the number of components has increased from 1 to 4.

2.13. Dominating sets

2.13.1. Definition and basic facts

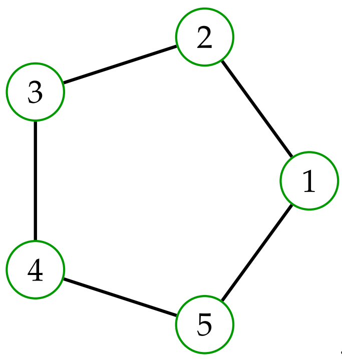
Here is another concept we can define for a graph:

Definition 2.13.1. Let $G = (V, E)$ be a simple graph.

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

A **dominating set** for G (or **dominating set** of G) will mean a subset of V that is dominating.

Example 2.13.2. Consider the cycle graph

$C_5 = (\{1,2,3,4,5\}, \{12, 23, 34, 45, 51\}) =$  .

[12]When we remove a vertex, we must of course also remove all edges that contain this vertex.

The set $\{1,3\}$ is a dominating set for C_5, since all three vertices $2,4,5$ that don't belong to $\{1,3\}$ have neighbors in $\{1,3\}$. The set $\{1,5\}$ is not a dominating set for C_5, since the vertex 3 has no neighbor in $\{1,5\}$. There is no dominating set for C_5 that has size 0 or 1, but there are several of size 2, and every subset of size ≥ 3 is dominating.

Here are some more examples:

- If $G = (V, E)$ is a simple graph, then the whole vertex set V is always dominating, whereas the empty set $\varnothing$ is dominating only when $V = \varnothing$.
- 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 (i.e., the more edges it has), the "easier" it is for a set to be dominating. Often, a graph is given, and one is interested in finding a dominating set of the smallest possible size[13]. As the case of an empty graph reveals, sometimes the only choice is the whole vertex set. However, in many cases, we can do better. Namely, we need to require that the graph has no isolated vertices:

Definition 2.13.3. Let G be a simple graph. A vertex v of G is said to be **isolated** if it has no neighbors (i.e., if $\deg v = 0$).

An isolated vertex has to belong to every dominating set (since otherwise, it would need a neighbor in that set, but it has no neighbors). Thus, isolated vertices do not contribute much to the study of dominating sets, other than inflating their size. Therefore, when we look for dominating sets, we can restrict ourselves to graphs with no isolated vertices. There, we have the following result:

Proposition 2.13.4. Let $G = (V, E)$ be a simple graph that has no isolated vertices. Then:

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

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

[13]Supposedly, this has applications in mobile networking: For example, you might want to choose a set of routers in a given network so that each node is either a router or directly connected (i.e., adjacent) to one.

One proof of this proposition will be given in Exercise 2.19 below (homework set #2 exercise 4). Another appears in [17s, §3.6].

For specific graphs, the bound $|V|/2$ in Proposition 2.13.4 **(a)** can often be improved. Here is an example:

Exercise 2.17. Let $n \geq 3$ be an integer. Find a formula for the smallest size of a dominating set of the cycle graph C_n. You can use the **ceiling function** $x \mapsto \lceil x \rceil$, which sends a real number x to the smallest integer that is $\geq x$.

Exercise 2.18. Let n and k be positive integers such that $n \geq k(k+1)$ and $k > 1$. Recall (from Subsection 2.6.3) the Kneser graph $KG_{n,k}$, whose vertices are the k-element subsets of $\{1, 2, \ldots, n\}$, and whose edges are the unordered pairs $\{A, B\}$ of such subsets with $A \cap B = \varnothing$.

Prove that the minimum size of a dominating set of $KG_{n,k}$ is $k+1$.

Exercise 2.19. Let $G = (V, E)$ be a connected simple graph with at least two vertices.

The **distance** $d(v, w)$ between two vertices v and w of G is defined to be the smallest length of a path from v to w. (In particular, $d(v, v) = 0$ for each $v \in V$.)

Fix a vertex $v \in V$. Define two subsets

$$A = \{w \in V \mid d(v, w) \text{ is even}\} \qquad \text{and} \qquad B = \{w \in V \mid d(v, w) \text{ is odd}\}$$

of V.

(a) Prove that A is dominating.

(b) Prove that B is dominating.

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

(d) Prove that the claim of part **(c)** holds even if we don't assume that G is connected, as long as we assume that each vertex of G has at least one neighbor. (In other words, prove Proposition 2.13.4 **(a)**.)

2.13.2. The number of dominating sets

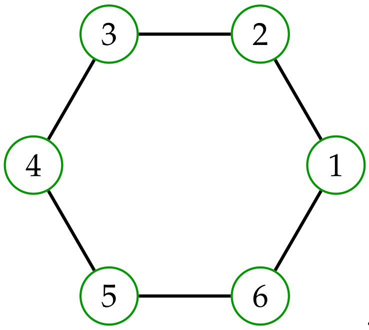
Next, we state a rather surprising recent result about the number of dominating sets of a graph:

Theorem 2.13.5 (Brouwer's dominating set theorem)**.** Let G be a simple graph. Then, the number of dominating sets of G is odd.

Three proofs of this theorem are given in Brouwer's note [Brouwe09].[14] Let me show the one I like the most. We first need a notation:

Definition 2.13.6. Let $G = (V, E)$ be a simple graph. A **detached pair** will mean a pair (A, B) of two disjoint subsets A and B of V such that there exists no edge $ab \in E$ with $a \in A$ and $b \in B$.

Example 2.13.7. Consider the cycle graph

$C_6 = (\{1,2,3,4,5,6\}, \{12, 23, 34, 45, 56, 61\}) =$  .

Then, $(\{1,2\}, \{4,5\})$ is a detached pair, whereas $(\{1,2\}, \{3,4\})$ is not (since 23 is an edge). Of course, there are many other detached pairs; in particular, any pair of the form $(\varnothing, B)$ or $(A, \varnothing)$ is detached.

Let me stress that the word "pair" always means "ordered pair" unless I say otherwise. So, if (A, B) is a detached pair, then (B, A) is a different detached pair, unless $A = B = \varnothing$.

Here is an attempt at a proof of Theorem 2.13.5. It is a nice example of how to apply known results to new graphs to obtain new results. The only problem is, it shows a result that is a bit at odds with the claim of the theorem...

Proof of Theorem 2.13.5, attempt 1. Write the graph G as (V, E).

Recall that $\mathcal{P}(V)$ denotes the set of all subsets of V.

Construct a new graph H with the vertex set $\mathcal{P}(V)$ as follows: Two subsets A and B of V are adjacent as vertices of H if and only if (A, B) is a detached pair. (Note that if the original graph G has n vertices, then this graph H has 2^n vertices. It is huge!)

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, the vertex A of H has odd degree if and only if A is a dominating set of G.

[14] Other proofs can be found in the AoPS thread `https://artofproblemsolving.com/community/c6h358772p1960068` . (This thread is concerned with a superficially different contest problem, but the latter problem is quickly revealed to be Theorem 2.13.5 in a number-theoretical disguise.)

[*Proof of Claim 1:* We let $N(A)$ denote the set of all vertices of G that have a neighbor in A. (This may or may not be disjoint from A.)

The neighbors of A (as a vertex in H) are precisely the subsets B of V such that (A, B) is a detached pair (by the definition of H). In other words, they are the subsets B of V that are disjoint from A and also have no neighbors in A (by the definition of a "detached pair"). In other words, they are the subsets B of V that are disjoint from A and also disjoint from $N(A)$. In other words, they are the subsets of the set $V \setminus (A \cup N(A))$. Hence, the number of such subsets B is $2^{|V \setminus (A \cup N(A))|}$.

The degree of A (as a vertex of H) is the number of neighbors of A in H. Thus, this degree is $2^{|V \setminus (A \cup N(A))|}$ (because we have just shown that the number of neighbors of A is $2^{|V \setminus (A \cup N(A))|}$). But 2^k is odd if and only if $k = 0$. Thus, we conclude that the degree of A (as a vertex of H) is odd if and only if $|V \setminus (A \cup N(A))| = 0$. The condition $|V \setminus (A \cup N(A))| = 0$ can be rewritten as follows:

$$\begin{aligned}
& (|V \setminus (A \cup N(A))| = 0) \\
\iff & (V \setminus (A \cup N(A)) = \varnothing) \\
\iff & (V \subseteq A \cup N(A)) \\
\iff & (V \setminus A \subseteq N(A)) \\
\iff & (\text{each vertex } v \in V \setminus A \text{ belongs to } N(A)) \\
\iff & (\text{each vertex } v \in V \setminus A \text{ has a neighbor in } A) \\
\iff & (A \text{ is dominating}) \qquad (\text{by the definition of "dominating"}).
\end{aligned}$$

Thus, what we have just shown is that the degree of A (as a vertex of H) is odd if and only if A is dominating. This proves Claim 1.]

Claim 1 shows that the vertices of H that have odd degree are precisely the dominating sets of G. But the handshake lemma (Corollary 2.4.4) tells us that any simple graph has an even number of vertices of odd degree. Applying this to H, we conclude that there is an even number of dominating sets of G.

Huh? We want to show that there is an **odd** number of dominating sets of G, not an even number! Why did we just get the opposite result?

Puzzle: Find the mistake in our above reasoning! The answer will be revealed on the next page. □

So what was the mistake in our reasoning?

The mistake is that our definition of H requires the vertex $\varnothing$ of H to be adjacent to itself (since $(\varnothing, \varnothing)$ is a detached pair); but a vertex of a simple graph cannot be adjacent to itself. So we need to tweak the definition of H somewhat:

Correction of the above proof of Theorem 2.13.5. Define the graph H as above, but do not try to have $\varnothing$ adjacent to itself. (This is the only vertex that creates any trouble, because a detached pair (A, B) cannot satisfy $A = B$ unless both A and B are $\varnothing$.)

We WLOG assume that $V \neq \varnothing$ (otherwise, the claim is obvious). Thus, the empty set $\varnothing$ is not dominating.

Our Claim 1 needs to be modified as follows:

> *Claim 1′:* Let A be a subset of V. Then, the vertex A of H has odd degree if and only if A is empty or a dominating set of G.

This can be proved in the same way as we "proved" Claim 1 above; we just need to treat the $A = \varnothing$ case separately now (but this case is easy: $\varnothing$ is adjacent to all other vertices of H, and thus has degree $2^{|V|} - 1$, which is odd).

So we conclude (using the handshake lemma) that the number of empty or dominating sets is even. Subtracting 1 for the empty set, we conclude that the number of dominating sets is odd (since the empty set is not dominating). This proves Brouwer's theorem (Theorem 2.13.5). $\square$

There are other ways to prove Brouwer's theorem as well. A particularly nice one was found by Irene Heinrich and Peter Tittmann in 2017; they gave an "explicit" formula for the number of dominating sets that shows that this number is odd ([HeiTit17, Theorem 8], restated using the language of detached pairs):

Theorem 2.13.8 (Heinrich–Tittmann formula)**.** Let $G = (V, E)$ be a simple graph with n vertices. Assume that $n > 0$.

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

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

Then:

(a) The numbers α and β are even.

(b) The number of dominating sets of G is $2^n - 1 + \alpha - \beta$.

Part **(a)** of this theorem is obvious (recall that if (A, B) is a detached pair, then so is (B, A)). Part **(b)** is the interesting part. In [17s, §3.3–§3.4], I give a long but elementary proof.

More recently ([HeiTit18]), Heinrich and Tittmann have refined their formula to allow counting dominating sets of a given size. Their main result is the following formula (exercise 5 on homework set #2):

Exercise 2.20. Let $G = (V, E)$ be a simple graph with at least one vertex. Let $n = |V|$. A **detached pair** means a pair (A, B) of two disjoint subsets A and B of V such that there exists no edge $ab \in E$ with $a \in A$ and $b \in B$.

Prove the following generalization of the Heinrich–Tittmann formula:

$$\sum_{\substack{S \text{ is a dominating} \\ \text{set of } G}} x^{|S|} = (1+x)^n - 1 + \sum_{\substack{(A,B) \text{ is a detached pair;} \\ A \neq \varnothing;\ B \neq \varnothing}} (-1)^{|A|} x^{|B|}.$$

(Here, both sides are polynomials in a single indeterminate x with coefficients in $\mathbb{Z}$.)

[**Hint:** This is a generalization of the Heinrich–Tittmann formula for the number of dominating sets. (The latter formula can be obtained fairly easily by substituting $x = 1$ into the above and subsequently cancelling the addends with $|A| \not\equiv |B| \bmod 2$ against each other.) You are free to copy arguments from [17s] and change whatever needs to be changed. (Some lemmas can even be used without any changes – they can then be cited without proof.)]

The following exercise gives a generalization of Theorem 2.13.5 (to recover Theorem 2.13.5 from it, set $k = 1$):

Exercise 2.21. Let k be a positive integer. Let $G = (V, E)$ be a simple graph. A subset U of V will be called ***k*-path-dominating** if for every $v \in V$, there exists a path of length $\leq k$ from v to some element of U.

Prove that the number of all k-path-dominating subsets of V is odd.

[**Hint:** This is not as substantial a generalization as it may look. The shortest proof is very short.]

[**Solution:** This is Exercise 6 on homework set #1 from my Spring 2017 course; see the course page for solutions.]

2.14. Hamiltonian paths and cycles

2.14.1. Basics

Now to something different. Here is a quick question: Given a simple graph G, when is there a closed **walk** that contains each vertex of G ?

The answer is easy: When G is connected. Indeed, if a simple graph G is connected, then we can label its vertices by $v_1, v_2, \ldots, v_n$ arbitrarily, and we then get a closed walk by composing a walk from v_1 to v_2 with a walk from v_2 to v_3 with a walk from v_3 to v_4 and so on, ending with a walk from v_n to

v_1. This closed walk will certainly contain each vertex. Conversely, such a walk cannot exist if G is not connected.

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

Definition 2.14.1. Let $G = (V, E)$ be a simple graph.

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

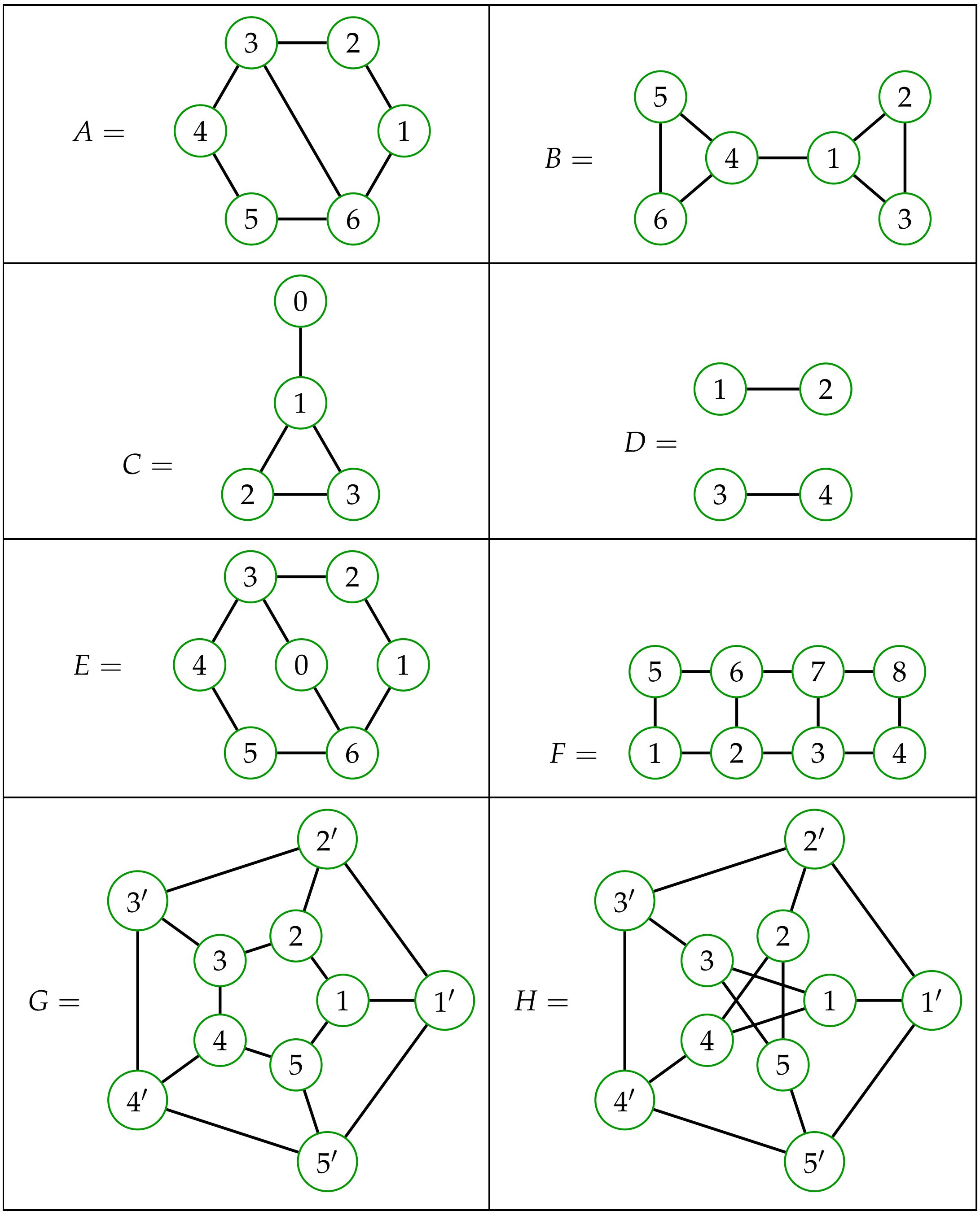
(b) A **Hamiltonian cycle** in G means a cycle $(v_0, v_1, \ldots, v_k)$ of G such that each vertex of G appears exactly once among $v_0, v_1, \ldots, v_{k-1}$.

Some graphs have Hamiltonian paths; some don't. Having a Hamiltonian cycle is even stronger than having a Hamiltonian path, because if $(v_0, v_1, \ldots, v_k)$ is a Hamiltonian cycle of G, then $(v_0, v_1, \ldots, v_{k-1})$ is a Hamiltonian path of G.

Convention 2.14.2. In the following, we will abbreviate:

- "Hamiltonian path" as "**hamp**";
- "Hamiltonian cycle" as "**hamc**".

Example 2.14.3. Which of the following eight graphs have hamps? Which have hamcs?



Answers:

- The graph A has a hamc $(1, 2, 3, 4, 5, 6, 1)$, and thus a hamp $(1, 2, 3, 4, 5, 6)$. (Recall that a graph that has a hamc always has a hamp, since we can simply remove the last vertex from a hamc to obtain a hamp.)

- The graph B has a hamp $(2,3,1,4,5,6)$, but no hamc. The easiest way to see that B has no hamc is the following: The edge 14 is a cut-edge (i.e., removing it renders the graph disconnected), thus a bridge (i.e., an edge that appears in no cycle); therefore, any cycle must stay entirely "on one side" of this edge.
- The graph C has a hamp $(0,1,2,3)$, but no hamc. The argument for the non-existence of a hamc is the same as for B: The edge 01 is a bridge.
- The graph D has neither a hamp nor a hamc, because it is not connected. Only a connected graph can have a hamp.
- The graph E has a hamp $(0,3,2,1,6,5,4)$, but no hamc (checking this requires some work, though).
- The graph F has a hamc $(1,2,3,4,8,7,6,5,1)$, thus also a hamp.
- The graph G has a hamc $(1,2,3,4,5,5',4',3',2',1',1)$, thus also a hamp.
- The graph H (which, by the way, is isomorphic to the Petersen graph from Subsection 2.6.3) has a hamp $(1,3,5,2,4,4',3',2',1',5')$, but no hamc (but this is not obvious! see the Wikipedia article for an argument).

In general, finding a hamp or a hamc, or proving that none exists, is a hard problem. It can always be solved by brute force (i.e., by trying all lists of distinct vertices and checking if there is a hamp among them, and likewise for hamcs), but this quickly becomes forbiddingly laborious as the size of the graph increases. Some faster algorithms exist (in particular, there is one of running time $O\left(n^2 2^n\right)$, where n is the number of vertices), but no polynomial-time algorithm is known. The problem (both in its hamp version and in its hamc version) is known to be NP-hard (in the language of complexity theory). In practice, hamps and hamcs can often be found with some wit and perseverance; proofs of their non-existence can often be obtained with some logic and case analysis (see the above example for some sample arguments). See the Wikipedia page for "Hamiltonian path problem" for more information.

The problem of finding hamps is related to the so-called "traveling salesman problem" (TSP), which asks for a hamp with "minimum weight" in a weighted graph (each edge has a number assigned to it, which is called its "weight", and the weight of a hamp is the sum of the weights of the edges it uses). There is a lot of computer-science literature about this problem.

2.14.2. Sufficient criteria: Ore and Dirac

We shall now show some necessary criteria and some sufficient criteria (but no necessary-and-sufficient criteria) for the existence of hamps and hamcs. Here is the most famous sufficient criterion:

Theorem 2.14.4 (Ore's theorem)**.** 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 vertices x and y. Then, G has a hamc.

There are various proofs of this theorem scattered around; see [Harju14, Theorem 3.6] or [Guicha16, Theorem 5.3.2]. We shall give another proof (following the "Algorithm" section on the Wikipedia page for "Ore's theorem"):

Proof of Theorem 2.14.4. A **listing** (of V) shall mean a list of elements of V that contains each element exactly once. It must clearly be an n-tuple.

The **hamness** of a listing $(v_1, v_2, \ldots, v_n)$ will mean the number of all $i \in \{1, 2, \ldots, n\}$ such that $v_i v_{i+1} \in E$. Here, we set $v_{n+1} = v_1$. (Visually, it is best to represent a listing $(v_1, v_2, \ldots, v_n)$ by drawing the vertices $v_1, v_2, \ldots, v_n$ on a circle in this order. Its hamness then counts how often two successive vertices on the circle are adjacent in the graph G.) Note that the hamness of a listing $(v_1, v_2, \ldots, v_n)$ does not change if we cyclically rotate the listing (i.e., transform it into $(v_2, v_3, \ldots, v_n, v_1)$).

Clearly, if we can find a listing $(v_1, v_2, \ldots, v_n)$ of hamness $\geq n$, then all of v_1v_2, v_2v_3, $\ldots$, v_nv_1 are edges of G, and thus $(v_1, v_2, \ldots, v_n, v_1)$ is a hamc of G. Thus, we need to find a listing of hamness $\geq n$.

To do so, I will show that if you have a listing of hamness $< n$, then you can slightly modify it to get a listing of larger hamness. In other words, I will show 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 the listing $(v_1, v_2, \ldots, v_n)$ has hamness $k < n$, there exists some $i \in \{1, 2, \ldots, n\}$ such that $v_i v_{i+1} \notin E$. Pick such an i. Thus, the vertices v_i and v_{i+1} of G are non-adjacent. The "$\deg x + \deg y \geq n$" assumption of the theorem thus yields $\deg(v_i) + \deg(v_{i+1}) \geq n$.

However,

$$\begin{aligned}
\deg(v_i) &= |\{w \in V \mid v_i w \in E\}| \\
&= \left|\{j \in \{1, 2, \ldots, n\} \mid v_i v_j \in E\}\right| \\
&= \left|\{j \in \{1, 2, \ldots, n\} \setminus \{i\} \mid v_i v_j \in E\}\right|
\end{aligned}$$

(because $j = i$ could not satisfy $v_i v_j \in E$ anyway) and

$$\begin{aligned}
\deg(v_{i+1}) &= |\{w \in V \mid v_{i+1}w \in E\}| \\
&= \left|\left\{j \in \{1, 2, \ldots, n\} \mid v_{i+1}v_{j+1} \in E\right\}\right| \\
&\qquad \left(\begin{array}{c}\text{since } (v_2, v_3, \ldots, v_{n+1}) \text{ is a listing of } V \\ (\text{because } v_{n+1} = v_1)\end{array}\right) \\
&= \left|\left\{j \in \{1, 2, \ldots, n\} \setminus \{i\} \mid v_{i+1}v_{j+1} \in E\right\}\right|
\end{aligned}$$

(because $j = i$ could not satisfy $v_{i+1}v_{j+1} \in E$ anyway). In light of these two equalities, we can rewrite the inequality $\deg(v_i) + \deg(v_{i+1}) \geq n$ as

$$\begin{aligned}
&\left|\left\{j \in \{1, 2, \ldots, n\} \setminus \{i\} \mid v_i v_j \in E\right\}\right| \\
&\qquad + \left|\left\{j \in \{1, 2, \ldots, n\} \setminus \{i\} \mid v_{i+1}v_{j+1} \in E\right\}\right| \geq n.
\end{aligned}$$

Thus, the two subsets $\left\{j \in \{1, 2, \ldots, n\} \setminus \{i\} \mid v_i v_j \in E\right\}$ and $\left\{j \in \{1, 2, \ldots, n\} \setminus \{i\} \mid v_{i+1}v_{j+1} \in E\right\}$ of the $(n-1)$-element set $\{1, 2, \ldots, n\} \setminus \{i\}$ have total size $\geq n$ (that is, the sum of their sizes is $\geq n$). Hence, these two subsets must overlap (i.e., have an element in common). In other words, there exists a $j \in \{1, 2, \ldots, n\} \setminus \{i\}$ that satisfies both $v_i v_j \in E$ and $v_{i+1}v_{j+1} \in E$. Pick such a j.

Now, consider a new listing obtained from the old listing $(v_1, v_2, \ldots, v_n)$ as follows:

- First, cyclically rotate the old listing so that it begins with v_{i+1}. Thus, you get the listing $(v_{i+1}, v_{i+2}, \ldots, v_n, v_1, v_2, \ldots, v_i)$.
- Then, reverse the part of the listing starting at v_{i+1} and ending at v_j. Thus, you get the new listing

$$\left(\underbrace{v_j, v_{j-1}, \ldots, v_{i+1}}_{\substack{\text{This is the reversed part;} \\ \text{it may or may not ``wrap around''} \\ \text{(i.e., contain } \ldots, v_1, v_n, \ldots \text{ somewhere).}}}, \underbrace{v_{j+1}, v_{j+2}, \ldots, v_i}_{\substack{\text{This is the part that} \\ \text{was not reversed.}}}\right).$$

  This is the new listing we want.

I claim that this new listing has hamness larger than k. Indeed, rotating the old listing clearly did not change its hamness. But reversing the part from v_{i+1} to v_j clearly did: After the reversal, the edges $v_i v_{i+1}$ and $v_j v_{j+1}$ no longer count towards the hamness (if they were edges to begin with), but the edges $v_i v_j$ and $v_{i+1}v_{j+1}$ started counting towards the hamness. This is a good bargain, because it means that the hamness gained $+2$ from the newly-counted edges $v_i v_j$ and

$v_{i+1}v_{j+1}$ (which, as we know, both exist), while only losing 0 or 1 (since the edge v_iv_{i+1} did not exist, whereas the edge v_jv_{j+1} may or may not have been lost). Thus, the hamness of the new listing is larger than the hamness of the old listing either by 1 or 2. In other words, it is larger than m by at least 1 or 2. This proves Claim 1.]

Now, we can start with **any** listing of V and keep modifying it using Claim 1, increasing its hamness each time, until its hamness becomes $\geq n$. But once its hamness is $\geq n$, we have found a hamc (as explained above). Theorem 2.14.4 is thus proven. $\square$

Corollary 2.14.5 (Dirac's theorem)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n \geq 3$.

Assume that $\deg x \geq \dfrac{n}{2}$ for each vertex $x \in V$.

Then, G has a hamc.

Proof. Follows from Ore's theorem, since any two vertices x and y of G satisfy $\underbrace{\deg x}_{\geq \frac{n}{2}} + \underbrace{\deg y}_{\geq \frac{n}{2}} \geq \dfrac{n}{2} + \dfrac{n}{2} = n$. $\square$

Exercise 2.22.

(a) Let $G = (V, E)$ be a simple graph, and let u and v be two distinct vertices of G that are not adjacent. Let $n = |V|$. Assume that $\deg u + \deg v \geq n$. Let $G' = (V, E \cup \{uv\})$ be the simple graph obtained from G by adding a new edge uv. Assume that G' has a hamc. Prove that G has a hamc.

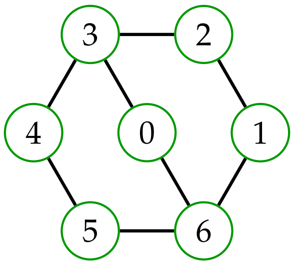
(b) Does this remain true if we replace "hamc" by "hamp"?

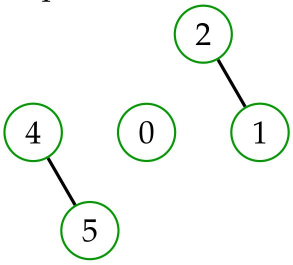
2.14.3. A necessary criterion

So much for sufficient criteria. What about necessary criteria?

Proposition 2.14.6. 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 $V \setminus S$. (In other words, this is the graph obtained from G by removing all vertices in S and removing all edges that have at least one endpoint in S.)

(For example, if $G =$ 

and $S = \{3, 6\}$, then $G \setminus S =$ 

.)

Also, we let $b_0(H)$ denote the number of connected components of a simple graph H.

(a) If G has a hamc, then every nonempty $S \subseteq V$ satisfies $b_0(G \setminus S) \leq |S|$.

(b) If G has a hamp, then every $S \subseteq V$ satisfies $b_0(G \setminus S) \leq |S| + 1$.

For example, part **(a)** of this proposition shows that the graph E from Example 2.14.3 has no hamc, because if we take S to be $\{3, 6\}$, then $b_0(G \setminus S) = 3$ whereas $|S| = 2$. Thus, the proposition can be used to rule out the existence of hamps and hamcs in some cases.

Proof of Proposition 2.14.6. **(a)** Let $S \subseteq V$ be a nonempty set. If we cut $|S|$ many vertices out of a cycle, then the cycle splits into at most $|S|$ paths:

Of course, our graph G itself may not be a cycle, but if it has a hamc, then the removal of the vertices in S will split the hamc into at most $|S|$ paths (according to the preceding sentence), and thus the graph $G \setminus S$ will have $\leq |S|$ many components (just using the surviving edges of the hamc alone). Taking into account all the other edges of G can only decrease the number of components.

(b) This is analogous to part **(a)**. $\square$

This proposition often (but not always) gives a quick way of convincing yourself that a graph has no hamc or hamp. Alas, its converse is false. Case in point: The Petersen graph (defined in Subsection 2.6.3) has no hamc, but it does satisfy the "every nonempty $S \subseteq V$ satisfies $b_0(G \setminus S) \leq |S|$" condition of Proposition 2.14.6 **(a)**.

2.14.4. Hypercubes

Now, let us move on to a concrete example of a graph that has a hamc.

Definition 2.14.7. Let $n \in \mathbb{N}$. The n-**hypercube** Q_n (more precisely, the n-**th hypercube graph**) is the simple graph with vertex set

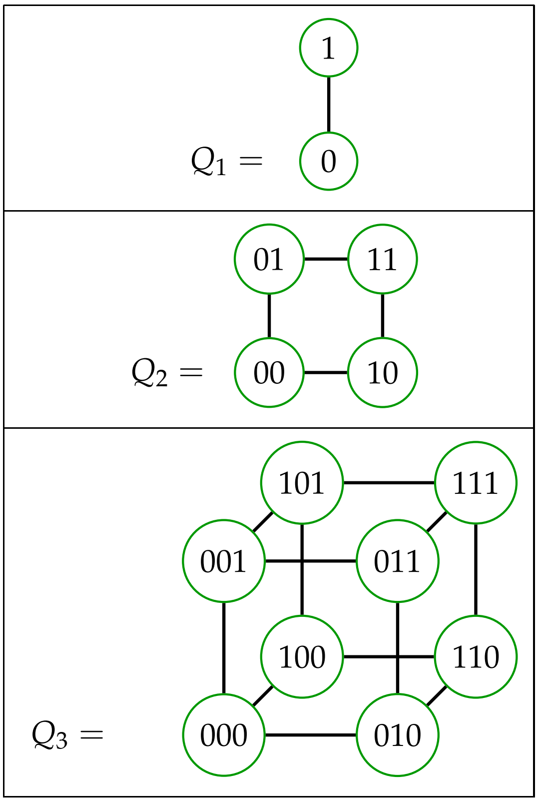
$$\{0,1\}^n = \{(a_1, a_2, \ldots, a_n) \mid \text{each } a_i \text{ belongs to } \{0,1\}\}$$

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

The elements of $\{0,1\}^n$ are often called **bitstrings** (or **binary words**), and 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 often write a bitstring $(a_1, a_2, \ldots, a_n)$ as $a_1 a_2 \cdots a_n$. (For example, we write $(0,1,1,0)$ as 0110.)

Example 2.14.8. Here is how the n-hypercubes Q_n look like for $n = 1, 2, 3$:



This should explain the name "hypercube". The 0-hypercube Q_0 is a graph with just one vertex (namely, the empty bitstring $()$).

Theorem 2.14.9 (Gray)**.** Let $n \geq 2$. Then, the graph Q_n has a hamc.

Such hamcs are known as **Gray codes**. They are circular lists of bitstrings of length n such that two consecutive bitstrings in the list always differ in exactly one bit. See the Wikipedia article on "Gray codes" for applications.

Proof of Theorem 2.14.9. We will show something stronger:

> *Claim 1:* For each $n \geq 1$, the n-hypercube Q_n has a hamp from $00 \cdots 0$ to $100 \cdots 0$.
>
> (Keep in mind that $00 \cdots 0$ and $100 \cdots 0$ are bitstrings, not numbers:
>
> $$00 \cdots 0 = \Big(\underbrace{0, 0, \ldots, 0}_{n \text{ zeroes}} \Big); \qquad 100 \cdots 0 = \Big(1, \underbrace{0, 0, \ldots, 0}_{n-1 \text{ zeroes}} \Big).$$
>
>)

[*Proof of Claim 1:* We induct on n.

Induction base: A look at Q_1 reveals a hamp from 0 to 1.

Induction step: Fix $N \geq 2$. We assume that Claim 1 holds for $n = N - 1$. In other words, Q_{N-1} has a hamp from $\underbrace{00\cdots0}_{N-1 \text{ zeroes}}$ to $1\underbrace{00\cdots0}_{N-2 \text{ zeroes}}$. Let $\mathbf{p}$ be such a hamp.

By attaching a 0 to the front of each bitstring (= vertex) in $\mathbf{p}$, we obtain a path $\mathbf{q}$ from $\underbrace{00\cdots0}_{N \text{ zeroes}}$ to $01\underbrace{00\cdots0}_{N-2 \text{ zeroes}}$ in Q_N.

By attaching a 1 to the front of each bitstring (= vertex) in $\mathbf{p}$, we obtain a path $\mathbf{r}$ from $1\underbrace{00\cdots0}_{N-1 \text{ zeroes}}$ to $11\underbrace{00\cdots0}_{N-2 \text{ zeroes}}$ in Q_N.

Now, we assemble a hamp from $\underbrace{00\cdots0}_{N \text{ zeroes}}$ to $1\underbrace{00\cdots0}_{N-1 \text{ zeroes}}$ in Q_N as follows:

- Start at $\underbrace{00\cdots0}_{N \text{ zeroes}}$, and follow the path $\mathbf{q}$ to its end (i.e., to $01\underbrace{00\cdots0}_{N-2 \text{ zeroes}}$).
- Then, move to the adjacent vertex $11\underbrace{00\cdots0}_{N-2 \text{ zeroes}}$.
- Then, follow the path $\mathbf{r}$ backwards, ending up at $1\underbrace{00\cdots0}_{N-1 \text{ zeroes}}$.

This shows that Claim 1 holds for $n = N$, too.]

Claim 1 tells us that the n-hypercube Q_n has a hamp from $00\cdots0$ to $100\cdots0$. Since its starting point $00\cdots0$ and its ending point $100\cdots0$ are adjacent, we can turn this hamp into a hamc by appending the starting point $00\cdots0$ again at the end. This proves Theorem 2.14.9. $\square$

2.14.5. Cartesian products

Theorem 2.14.9 can in fact be generalized. To state the generalization, we define the **Cartesian product** of two graphs:

Definition 2.14.10. Let $G = (V, E)$ and $H = (W, F)$ be two simple graphs. The **Cartesian product** $G \times H$ of these two graphs is defined to be the simple graph $(V \times W,\ E' \cup F')$, where

$$E' := \{(v_1, w)(v_2, w) \mid v_1v_2 \in E \text{ and } w \in W\} \qquad \text{and}$$
$$F' := \{(v, w_1)(v, w_2) \mid w_1w_2 \in F \text{ and } v \in V\}.$$

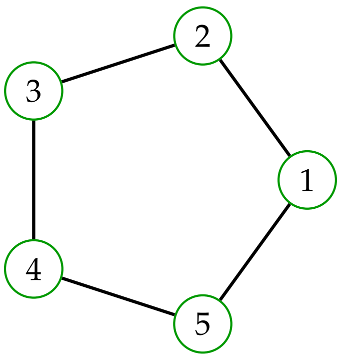
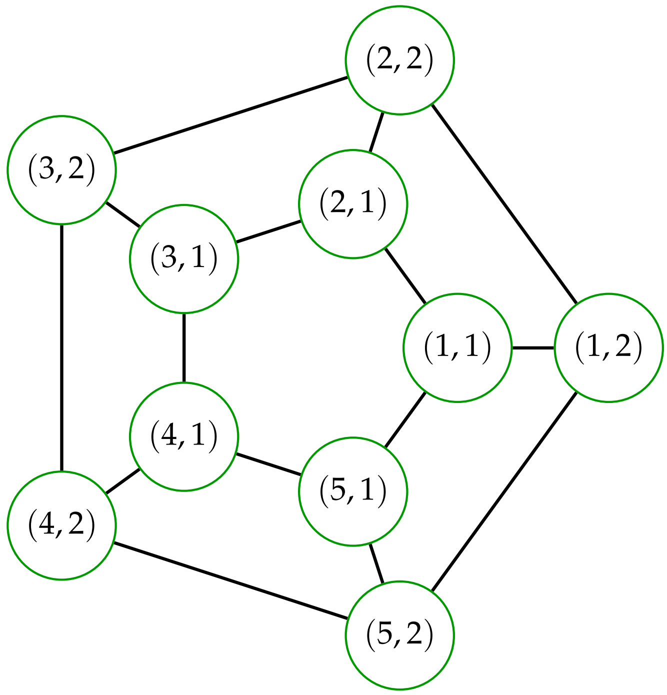
In other words, it is the graph whose vertices are pairs $(v, w) \in V \times W$ consisting of a vertex of G and a vertex of H, and whose edges are of the forms

$$(v_1, w)(v_2, w) \qquad \text{where } v_1v_2 \in E \text{ and } w \in W$$

and

$$(v, w_1)\,(v, w_2) \qquad \text{where } w_1 w_2 \in F \text{ and } v \in V.$$

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 to the corresponding vertex of the second copy by an edge. (The vertices of the first copy are the $(v, 1)$, whereas the vertices of the second copy are the $(v, 2)$.) For a specific example, here is the 5-cycle graph C_5 and the Cartesian product $C_5 \times P_2$:

C_5	$C_5 \times P_2$
	

As another instance of the above description of $G \times P_2$, it is easy to see the following:

Proposition 2.14.11. We have $Q_n \cong Q_{n-1} \times P_2$ for each $n \geq 1$. (See Definition 2.14.7 for the definitions of Q_n and Q_{n-1}.)

Proof. This is Exercise 1 **(a)** on homework set #2 from my Spring 2017 course; see the course page for solutions. □

Now, we claim the following:

Theorem 2.14.12. 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) Now assume furthermore that at least one of the two numbers $|\mathrm{V}(G)|$ and $|\mathrm{V}(H)|$ is even, and that both numbers $|\mathrm{V}(G)|$ and $|\mathrm{V}(H)|$ are larger than 1. Then, the Cartesian product $G \times H$ has a hamc.

Proof. This is Exercise 1 on homework set #2 from my Spring 2017 course (specifically, its parts **(b)** and **(c)**). Its solution can be found on the course page. (Specifically, see the solution to Exercise 1 on homework set #2 from Spring 2017.) $\square$

Now, Theorem 2.14.9 can be reproved (again by inducting on n) using Theorem 2.14.12 **(b)** and Proposition 2.14.11, since P_2 has a hamp and since $|\mathrm{V}(P_2)| = 2$ is even. (Convince yourself that this works!)

2.14.6. Subset graphs

The n-hypercube Q_n can be reinterpreted in terms of subsets of $\{1, 2, \ldots, n\}$. Namely: Let $n \in \mathbb{N}$. Let G_n be the simple graph whose vertex set is the powerset $\mathcal{P}(\{1, 2, \ldots, n\})$ of $\{1, 2, \ldots, n\}$ (that is, the vertices are all 2^n subsets of $\{1, 2, \ldots, n\}$), and whose edges are determined as follows: Two vertices S and T are adjacent if and only if one of the two sets S and T is obtained from the other by inserting an extra element (i.e., we have either $S = T \cup \{s\}$ for some $s \notin T$, or $T = S \cup \{t\}$ for some $t \notin S$). Then, $G_n \cong Q_n$, since the map

$$\{0,1\}^n \to \mathcal{P}(\{1, 2, \ldots, n\}),$$
$$(a_1, a_2, \ldots, a_n) \mapsto \{i \in \{1, 2, \ldots, n\} \mid a_i = 1\}$$

is a graph isomorphism from Q_n to G_n.

Thus, Theorem 2.14.9 shows that for each $n \geq 2$, the graph G_n has a hamc. In other words, for each $n \geq 2$, we can list all subsets of $\{1, 2, \ldots, n\}$ in a circular list in such a way that each subset on this list is obtained from the previous one by inserting or removing a single element. For example, for $n = 3$, here is such a list:

$$\varnothing, \ \{1\}, \ \{1,2\}, \ \{2\}, \ \{2,3\}, \ \{1,2,3\}, \ \{1,3\}, \ \{3\}.$$

A long-standing question only resolved a few years ago asked whether the same can be done with the subsets of $\{1, 2, \ldots, n\}$ having size $\dfrac{n \pm 1}{2}$ when n is odd. For example, for $n = 3$, we can do it as follows:

$$\{1\}, \ \{1,2\}, \ \{2\}, \ \{2,3\}, \ \{3\}, \ \{1,3\}.$$

In other words, if $n \geq 3$ is odd, and if G'_n is the induced subgraph of G_n on the set of all subsets J of $\{1, 2, \ldots, n\}$ that satisfy $|J| \in \left\{ \dfrac{n-1}{2}, \dfrac{n+1}{2} \right\}$, then does G'_n have a hamc?

Since $G_n \cong Q_n$, we can restate this question equivalently as follows: If $n \geq 3$ is odd, and if Q'_n is the induced subgraph of Q_n on the set

$$\left\{ a_1 a_2 \cdots a_n \in \{0,1\}^n \mid \sum_{i=1}^{n} a_i \in \left\{ \frac{n-1}{2}, \frac{n+1}{2} \right\} \right\},$$

then does Q'_n have a hamc?

In 2014, Torsten Mütze proved that the answer is "yes". See [Mutze14] for his truly nontrivial proof, and [Mutze22] for a recent survey of similar questions. (Cf. also change ringing.)

The following exercise provides another generalization of Theorem 2.14.9:

Exercise 2.23. Let n and k be two integers such that $n > k > 0$. Define the simple graph $Q_{n,k}$ as follows: Its vertices are the bitstrings $(a_1, a_2, \ldots, a_n) \in \{0,1\}^n$; two such bitstrings are adjacent if and only if they differ in exactly k bits (in other words: two vertices $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ are adjacent if and only if the number of $i \in \{1, 2, \ldots, n\}$ satisfying $a_i \neq b_i$ equals k). (Thus, $Q_{n,1}$ is the n-hypercube graph Q_n.)

(a) Does $Q_{n,k}$ have a hamc when k is even? (Recall that "hamc" is short for "Hamiltonian cycle".)

(b) Does $Q_{n,k}$ have a hamc when k is odd?

[**Hint:** One way to approach part **(b)** is by identifying the set $\{0,1\}$ with the field $\mathbb{F}_2$ with two elements. The bitstrings $(a_1, a_2, \ldots, a_n) \in \{0,1\}^n$ thus become the size-n row vectors in the $\mathbb{F}_2$-vector space $\mathbb{F}_2^n$. Let $e_1, e_2, \ldots, e_n$ be the standard basis vectors of $\mathbb{F}_2^n$ (so that e_i has a 1 in its i-th position and zeroes everywhere else). Then, two vectors are adjacent in the n-hypercube graph Q_n (resp. in the graph $Q_{n,k}$) if and only if their difference is one of the standard basis vectors (resp., a sum of k distinct standard basis vectors). Try to use this to find a graph isomorphism from Q_n to a subgraph of $Q_{n,k}$.]

The next exercise extends the idea of our proof of Theorem 2.14.9:

Exercise 2.24. Let $n \geq 1$. Let Q_n be the n-hypercube graph, as in Definition 2.14.7. Recall that "hamp" is short for "Hamiltonian path".

At what vertices can a hamp of Q_n end if it starts at the vertex $00 \cdots 0$? (Find all possibilities, and prove that they are possible and all other vertices are impossible.)

3. Multigraphs

3.1. Definitions

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

Definition 3.1.1. 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. In other words,

$$\mathcal{P}_{1,2}(V) := \{S \subseteq V \mid |S| \in \{1,2\}\}$$
$$= \{\{u,v\} \mid u,v \in V \text{ not necessarily distinct}\}.$$

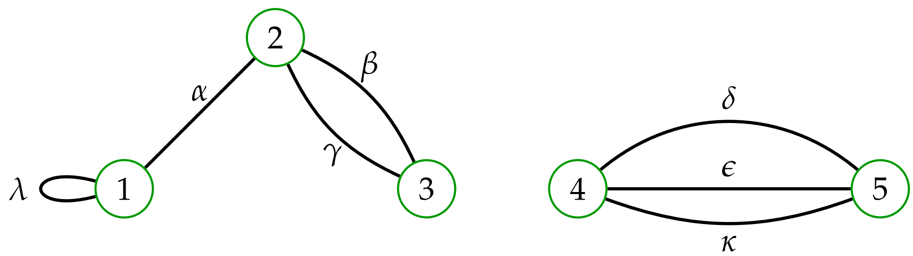
For instance,

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

We can now define multigraphs:

Definition 3.1.2. 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 3.1.3. Here is a multigraph:



Formally speaking, this multigraph is the triple (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 that sends $\alpha,\beta,\gamma,\delta,\varepsilon,\kappa,\lambda$ to $\{1,2\}, \{2,3\}, \{2,3\}, \{4,5\}, \{4,5\}, \{4,5\}, \{1\}$, respectively. (Of course, you can write $\{1\}$ as $\{1,1\}$.)

This suggests the following terminology (most of which is a calque of our previously defined terminology for simple graphs):

Definition 3.1.4. Let $G = (V, E, \varphi)$ be a multigraph. Then:

(a) The elements of V are called the **vertices** of G.

The set V is called the **vertex set** of G, and is denoted $\operatorname{V}(G)$.

(b) The elements of E are called the **edges** of G.

The set E is called the **edge set** of G, and is denoted $\operatorname{E}(G)$.

(c) If e is an edge of G, then the elements of $\varphi(e)$ are called the **endpoints** of e.

(d) We say that an edge e **contains** a vertex v if $v \in \varphi(e)$ (in other words, if v is an endpoint of e).

(e) Two vertices u and v are said to be **adjacent** if there exists an edge $e \in E$ whose endpoints are u and v.

(f) Two edges e and f are said to be **parallel** if $\varphi(e) = \varphi(f)$. (In the above example, any two of the edges $\delta, \varepsilon, \kappa$ are parallel.)

(g) We say that G has **no parallel edges** if G has no two distinct edges that are parallel.

(h) 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). (In Example 3.1.3, the edge λ is a loop.)

(i) We say that G is **loopless** if G has no loops (among its edges).

(j) 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,

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

(Note that, unlike in the case of a simple graph, $\deg v$ is **not** the number of neighbors of v, unless it happens that v is not contained in any loops or parallel edges.)

(For example, in Example 3.1.3, we have $\deg 1 = 3$ and $\deg 2 = 3$ and $\deg 3 = 2$ and $\deg 4 = 3$ and $\deg 5 = 3$.)

(k) 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 each edge e_i are v_{i-1} and v_i). Note that we have to record both the vertices **and** the edges in our walk, since we want the walk to "know" which edges it traverses. (For instance, in Example 3.1.3, the two walks $(1, \alpha, 2, \beta, 3)$ and $(1, \alpha, 2, \gamma, 3)$ are distinct.)

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 $e_1, e_2, \ldots, e_k$. This walk is said to **start** at v_0 and **end** at v_k; it is also said to be a **walk from** v_0 **to** v_k. Its **starting point** is v_0, and its **ending point** is v_k. Its **length** is k.

(l) A **path** means a walk whose vertices are distinct.

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

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

(o) 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, as we did for simple graphs. Thus, in Example 3.1.3, both $(2, \beta, 3, \gamma, 2)$ and $(1, \lambda, 1)$ are cycles, but $(2, \beta, 3, \beta, 2)$ is not. The purpose of the "$k \geq 3$" requirement for cycles in simple graphs was to disallow closed walks such as $(2, \beta, 3, \beta, 2)$ from being cycles; but they are now excluded by the "the edges $e_1, e_2, \ldots, e_k$ are distinct" condition.)

(p) Hamiltonian paths and cycles are defined as for simple graphs.

(q) We draw a multigraph by drawing each vertex as a point, each edge as a curve, and labeling both the vertices and the edges (or not, if we don't care about what they are). An example of such a drawing appeared in Example 3.1.3.

So there are 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 two equal ones, 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 much in common; thus, they are both called "graphs":

Convention 3.1.5. The word "**graph**" means either "simple graph" or "multigraph". The precise meaning should usually be understood from the context. (I will try not to use it when it could cause confusion.)

Fortunately, simple graphs and multigraphs have many properties in common, and often it is not hard to derive a result about multigraphs from the analogous result about simple graphs or vice versa. We will soon explore how some of the properties we have seen in the previous chapter can be adapted to multigraphs. First, however, let us explain how to convert multigraphs into simple graphs and vice versa.

3.2. Conversions

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

Definition 3.2.1. 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 u and v are adjacent in G. Thus, G^{simp} is obtained from G by removing loops and "collapsing" parallel edges to a single edge.

For example, the underlying simple graph of the multigraph G in Example 3.1.3 is

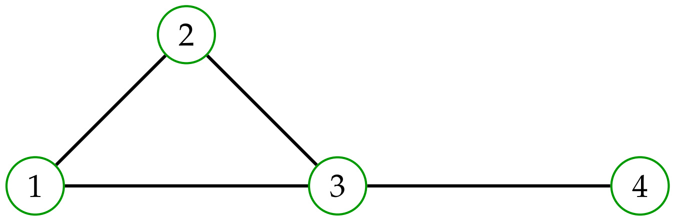
Conversely, each simple graph can be viewed as a multigraph:

Definition 3.2.2. Let $G = (V, E)$ be a simple graph. Then, the **corresponding multigraph** G^{mult} is defined to be the multigraph

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

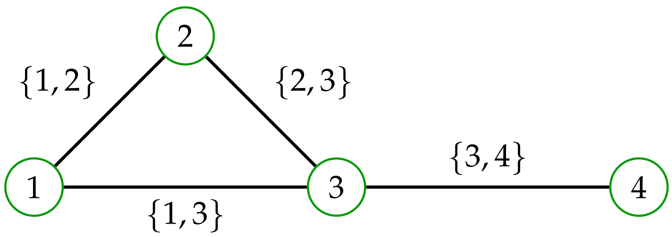
where $\iota : E \to \mathcal{P}_{1,2}(V)$ is the map sending each $e \in E$ to e itself.

Example 3.2.3. If

$G =$ 

,

then

$G^{\text{mult}} =$ 

.

As we said, the "underlying simple graph" construction $G \mapsto G^{\text{simp}}$ destroys information, so it is irreversible. This being said, the two constructions $G \mapsto G^{\text{simp}}$ and $G \mapsto G^{\text{mult}}$ come fairly close to undoing one another:[15]

Proposition 3.2.4.

(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 $\left(G^{\text{simp}}\right)^{\text{mult}} \cong G$. (This is just an isomorphism, not an equality, since the "identities" of the edges of G have been forgotten in G^{simp} and cannot be recovered.)

(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 and thus is not isomorphic to G.

[15] In the following proposition, we will use the notion of an "isomorphism of multigraphs". A rigorous definition of this notion is given in Definition 3.3.4 further below (but it is more or less what you would expect: it is a way to relabel the vertices and the edges of one multigraph to obtain those of another).

Proof. A matter of understanding the definitions. □

We will often identify a simple graph G with the corresponding multigraph G^{mult}. This may be dangerous, because we have defined notions such as adjacency, walks, paths, cycles, etc. both for simple graphs and for multigraphs; thus, when we identify a simple graph G with the multigraph G^{mult}, we are potentially inviting ambiguity (for example, does "cycle of G" mean a cycle of the simple graph G or of the multigraph G^{mult} ?). Fortunately, this ambiguity is harmless, because whenever G is a simple graph, any of the notions we defined for G is equivalent to the corresponding notion for the multigraph G^{mult}. For example, for the notions of a cycle, we have the following:

Proposition 3.2.5. 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. This is not completely obvious, since our definitions of a cycle of a simple graph and of a cycle of a multigraph were somewhat different. The proof boils down to checking the following two statements:

1. If $(v_0, v_1, \ldots, v_k)$ is a cycle of the simple graph G, then its edges $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}$ are distinct.
2. 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 $k \geq 3$.

Checking statement 2 is easy (we cannot have $k = 1$ since G^{mult} has no loops, and we cannot have $k = 2$ since this would lead to $e_1 = e_2$). Statement 1 is also clear, since the distinctness of the k vertices $v_0, v_1, \ldots, v_{k-1}$ forces the 2-element sets formed from these k vertices to also be distinct (and since the edges $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\} = \{v_{k-1}, v_0\}$ are such 2-element sets). □

For all other notions discussed above, it is even more obvious that there is no ambiguity.

3.3. Generalizing from simple graphs to multigraphs

Now, as promised, we shall revisit the results of Chapter 2, and see which of them also hold for multigraphs instead of simple graphs.

3.3.1. The Ramsey number $R(3,3)$

One of the first properties of simple graphs that we proved is the following (Proposition 2.3.1):

> **Proposition 3.3.1.** Let G be a simple graph with $|\mathrm{V}(G)| \geq 6$ (that is, G has at least 6 vertices). Then, at least one of the following two statements holds:
>
> - *Statement 1:* There exist three distinct vertices a, b and c of G such that ab, bc and ca are edges of G.
> - *Statement 2:* There exist three distinct vertices a, b and c of G such that none of ab, bc and ca is an edge of G.

This is still true for multigraphs[16], because replacing a multigraph G by the underlying simple graph G^{simp} does not change the meaning of the statement.

3.3.2. Degrees

In Definition 2.4.1, we defined the degree of a vertex v in a simple graph $G = (V, E)$ by

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

These equalities **no longer hold** when G is a multigraph. Parallel edges correspond to the same neighbor, so the number of neighbors of v is only a lower bound on $\deg v$.

Proposition 2.4.2 (which says that if G is a simple graph with n vertices, then any vertex v of G has degree $\deg v \in \{0, 1, \ldots, n-1\}$) also **no longer holds** for multigraphs, because you can have arbitrarily many edges in a multigraph with just 1 or 2 vertices. (You can even have parallel loops!)

Is Proposition 2.4.3 true for multigraphs? Yes, because we have said that loops should count twice in the definition of the degree. The proof needs some tweaking, though. Let me give a slightly different proof; but first, let me state the claim for multigraphs as a proposition of its own:

[16] Of course, we should understand it appropriately: i.e., we should read "*ab* is an edge" as "there is an edge with endpoints *a* and *b*".

Proposition 3.3.2 (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. Write G as $G = (V, E, \varphi)$; thus, $\operatorname{V}(G) = V$ and $\operatorname{E}(G) = E$.

For each edge e, let us (arbitrarily) choose one endpoint of e and denote it by $\alpha(e)$. The other endpoint will be called $\beta(e)$. If e is a loop, then we set $\beta(e) = \alpha(e)$. Then, for each vertex v, we have

$$\begin{aligned} \deg v = {} & (\text{the number of } e \in E \text{ such that } v = \alpha(e)) \\ & + (\text{the number of } e \in E \text{ such that } v = \beta(e)) \end{aligned}$$

(note how loops get counted twice on the right hand side, because if $e \in E$ is a loop, then v is both $\alpha(e)$ and $\beta(e)$ at the same time). Summing up this equality over all $v \in V$, we obtain

$$\begin{aligned} \sum_{v \in V} \deg v = {} & \sum_{v \in V} (\text{the number of } e \in E \text{ such that } v = \alpha(e)) \\ & + \sum_{v \in V} (\text{the number of } e \in E \text{ such that } v = \beta(e)) . \end{aligned}$$

However,

$$\sum_{v \in V} (\text{the number of } e \in E \text{ such that } v = \alpha(e)) = |E| ,$$

since each edge $e \in E$ is counted in exactly one addend of this sum. Similarly,

$$\sum_{v \in V} (\text{the number of } e \in E \text{ such that } v = \beta(e)) = |E| .$$

Thus, the above equality becomes

$$\begin{aligned} \sum_{v \in V} \deg v = {} & \underbrace{\sum_{v \in V} (\text{the number of } e \in E \text{ such that } v = \alpha(e))}_{=|E|} \\ & + \underbrace{\sum_{v \in V} (\text{the number of } e \in E \text{ such that } v = \beta(e))}_{=|E|} \\ = {} & |E| + |E| = 2 \cdot |E| . \end{aligned}$$

This proves Proposition 3.3.2. □

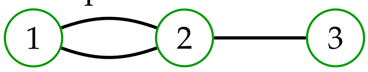
This is a good motivation for counting loops twice in the definition of a degree.

The handshake lemma (Corollary 2.4.4) **still holds for multigraphs**. In other words, we have the following:

Corollary 3.3.3 (handshake lemma)**.** Let G be a multigraph. Then, the number of vertices v of G whose degree $\deg v$ is odd is even.

Proof. This follows from Proposition 3.3.2 in the same way as for simple graphs. $\square$

Proposition 2.4.5 **fails for multigraphs**. For example, the multigraph



has three vertices with degrees $1, 2, 3$. Fortunately, Proposition 2.4.5 was more of a curiosity than a useful fact.

Mantel's theorem (Theorem 2.4.6) also **fails for multigraphs**, because we can join two vertices with a lot of parallel edges and thus satisfy $e > n^2/4$ for stupid reasons without ever creating a triangle. Thus, Turan's theorem (Theorem 2.4.8) also fails for multigraphs.

3.3.3. Graph isomorphisms

Graph isomorphy (and isomorphisms) can still be defined for multigraphs, but the definition is not the same as for simple graphs. Graph isomorphisms can no longer be defined merely as bijections between the vertex sets, since we also need to specify what they do to the edges. Instead, we define them as follows:

Definition 3.3.4. 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 $e \in E$, then the endpoints of $\beta(e)$ are the images under α of the endpoints of e. (This property can also be restated as a commutative diagram

$$\begin{tikzcd} E \arrow[r, "\beta"] \arrow[d, "\varphi"'] & F \arrow[d, "\psi"] \\ \mathcal{P}_{1,2}(V) \arrow[r, "\mathcal{P}(\alpha)"'] & \mathcal{P}_{1,2}(W) \end{tikzcd},$$

where $\mathcal{P}(\alpha)$ is the map from $\mathcal{P}_{1,2}(V)$ to $\mathcal{P}_{1,2}(W)$ that sends each subset $\{u, v\} \in \mathcal{P}_{1,2}(V)$ to $\{\alpha(u), \alpha(v)\} \in \mathcal{P}_{1,2}(W)$. If you are used to category theory, this restatement may look more natural to you.)

(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.

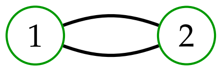
Again, isomorphy of multigraphs is an equivalence relation.

3.3.4. Complete graphs, paths, cycles

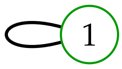
In Definition 2.6.1, Definition 2.6.2 and Definition 2.6.3, we defined the complete graphs K_n, the path graphs P_n and the cycle graphs C_n as simple graphs. Thus, all of them can be viewed as multigraphs if one so desires (since each simple graph G gives rise to a multigraph G^{mult}).

However, using multigraphs, we can extend our definition of n-th cycle graphs C_n to the case $n = 1$ and also tweak it in the case $n = 2$ to make it more natural. We do this as follows:

Definition 3.3.5. We modify the definition of cycle graphs (Definition 2.6.3) as follows:

(a) We **redefine** the 2-nd cycle graph C_2 to be the multigraph with two vertices 1 and 2 and two parallel edges with endpoints 1 and 2. (We don't care what the edges are, only that there are two of them and each has endpoints 1 and 2.) Thus, it looks as follows: 

.

(b) We define the 1-st cycle graph C_1 to be the multigraph with one vertex 1 and one edge (which is necessarily a loop). Thus, it looks as follows: 

.

This has the effect that the n-th cycle graph C_n has exactly n edges for each $n \geq 1$ (rather than having 1 edge for $n = 2$, as it did back when it was a simple graph).

3.3.5. Induced submultigraphs

In Definition 2.7.1, we defined subgraphs and induced subgraphs of a simple graph. The corresponding notions for multigraphs are defined as follows:

Definition 3.3.6. Let $G = (V, E, \varphi)$ be a multigraph.

(a) A **submultigraph** of G means a multigraph of the form $H = (W, F, \psi)$, where $W \subseteq V$ and $F \subseteq E$ and $\psi = \varphi\mid_F$. In other words, a submultigraph of G means a multigraph H whose vertices are vertices of G and whose edges are edges of G and whose edges have the same endpoints in H as they do in G.

We often abbreviate "submultigraph" as "**subgraph**".

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

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

of G, where

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

In other words, it denotes the submultigraph of G whose vertices are the elements of S, and whose edges are precisely those edges of G whose both endpoints belong to S. We denote this induced submultigraph by $G[S]$.

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

The infix "multi" is often omitted. So we often speak of "subgraphs" instead of "submultigraphs".

With these definitions, we can now identify cycles in a multigraph with subgraphs isomorphic to a cycle graph: A cycle of length n in a multigraph G is "the same as" a submultigraph of G isomorphic to C_n. (We leave the details to the reader.)

3.3.6. Disjoint unions

In Section 2.8, we defined the disjoint union of two or more simple graphs. The analogous definition for multigraphs is straightforward and left to the reader.

3.3.7. Walks

We already defined walks, paths, closed walks and cycles for multigraphs back in Section 3.1. The **length** of a walk is still defined to be its number of edges. Now, let's see which of their basic properties (seen in Section 2.9) still hold for multigraphs.

First of all, the edges of a path are still always distinct. This is just as easy to prove as for simple graphs.

Next, let us see how two walks can be "spliced" together:

Proposition 3.3.7. Let G be a multigraph. Let u, v and w be three vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ be a walk from u to v. Let

$\mathbf{b} = (b_0, f_1, b_1, \ldots, f_\ell, b_\ell)$ be a walk from v to w. Then,

$$\begin{aligned} &(a_0, e_1, a_1, \ldots, e_k, a_k, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \\ &= (a_0, e_1, a_1, \ldots, a_{k-1}, e_k, b_0, f_1, b_1, \ldots, f_\ell, b_\ell) \\ &= (a_0, e_1, a_1, \ldots, a_{k-1}, e_k, v, f_1, b_1, \ldots, f_\ell, b_\ell) \end{aligned}$$

is a walk from u to w. This walk shall be denoted $\mathbf{a} * \mathbf{b}$.

Walks can be reversed (i.e., walked in backwards direction):

Proposition 3.3.8. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ be a walk from u to v. Then:

(a) The list $(a_k, e_k, a_{k-1}, e_{k-1}, \ldots, e_1, a_0)$ is a walk from v to u. We denote this walk by $\operatorname{rev} \mathbf{a}$ and call it the **reversal** of $\mathbf{a}$.

(b) If $\mathbf{a}$ is a path, then $\operatorname{rev} \mathbf{a}$ is a path again.

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

Proposition 3.3.9. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ 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.

Corollary 3.3.10 (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 $\leq k$.

All these results can be proved in the same way as their counterparts for simple graphs; the only change needed is to record the edges in the walk.

Given a multigraph G and two vertices u and v of G, we can ask ourselves the same five Questions 1, 2, 3, 4 and 5 that we asked for a simple graph G in Subsection 2.9.4. The answers we gave in that subsection still apply without requiring substantial changes; the only necessary modification is that we now have to keep track of the edges in a path or walk. (The reader can easily fill in the details here.)

3.3.8. Path-connectedness

The relation "path-connected" is defined for multigraphs just as it is for simple graphs (Definition 2.9.8), and is still denoted $\simeq_G$. It is still an equivalence relation (and the proof is the same as for simple graphs). The following also holds (with the same proof as for simple graphs):

Proposition 3.3.11. Let G be a multigraph. Let u and v be two vertices of G. Then, $u \simeq_G v$ if and only if there exists a path from u to v.

The definitions of "components" and "connected" for multigraphs are the same as for simple graphs (Definition 2.9.11 and Definition 2.9.12). The following propositions can be proved in the same way as we proved their analogues for simple graphs:

Proposition 3.3.12. Let G be a multigraph. Let C be a component of G.

Then, the induced subgraph (= submultigraph) of G on the set C is connected.

Proposition 3.3.13. Let G be a multigraph. Let $C_1, C_2, \ldots, C_k$ be all components of G (listed without repetition).

Thus, G is isomorphic to the disjoint union $G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k]$.

The following proposition is an analogue of Proposition 2.10.4 for multigraphs:

Proposition 3.3.14. Let G be a multigraph. Let $\mathbf{w}$ be a walk of G such that no two adjacent edges of $\mathbf{w}$ are identical. (By "adjacent edges", we mean edges of the form e_{i-1} and e_i, where $e_1, e_2, \ldots, e_k$ are the edges of $\mathbf{w}$ from first to last.)

Then, $\mathbf{w}$ either is a path or contains a cycle (i.e., there exists a cycle of G whose edges are edges of $\mathbf{w}$).

Proof. The proof of this proposition for multigraphs is more or less the same as it was for simple graphs (i.e., as the proof of Proposition 2.10.4), with a mild difference in how we prove that the walk $(w_i, w_{i+1}, \ldots, w_j)$ is a cycle (of course, this walk is no longer $(w_i, w_{i+1}, \ldots, w_j)$ now, but rather $(w_i, e_{i+1}, w_{i+1}, \ldots, e_j, w_j)$, because the edges need to be included).[17] $\square$

[17]Here are some details:

We assume that $\mathbf{w}$ is not a path, and we write the walk $\mathbf{w}$ as $(w_0, e_1, w_1, e_2, w_2, \ldots, w_k, e_k)$. Then, there exists a pair (i, j) of integers i and j with $i < j$ and $w_i = w_j$. Among all such pairs, we pick one with **minimum** difference $j - i$. Then, $(w_i, e_{i+1}, w_{i+1}, \ldots, e_j, w_j)$ is a closed walk. We claim that this closed walk is a cycle.

To do so, we need to show that

1. the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct;
2. the edges $e_{i+1}, e_{i+2}, \ldots, e_j$ are distinct;
3. we have $j - i \geq 1$.

The first of these claims follows from the minimality of $j - i$. The third follows from $i < j$. It remains to prove the second claim. In other words, it remains to prove that the edges $e_{i+1}, e_{i+2}, \ldots, e_j$ are distinct, i.e., that we have $e_a \neq e_b$ for any two integers a and b satisfying

Just as for simple graphs, we get the following corollary:

Corollary 3.3.15. Let G be a multigraph. Assume that G has a closed walk $\mathbf{w}$ of length > 0 such that no two adjacent edges of $\mathbf{w}$ are identical. Then, G has a cycle.

The analogue of Theorem 2.10.7 for multigraphs is true as well:

Theorem 3.3.16. Let G be a multigraph. Let u and v be two vertices in G. Assume that there are two distinct paths from u to v. Then, G has a cycle.

Proof. For simple graphs, this was proved as Theorem 2.10.7 above. The same proof applies to multigraphs, once the obvious changes are made (e.g., instead of $p_{a-1}p_a$ and $q_{b-1}q_b$, we need to take the last edges of the two walks $\mathbf{p}$ and $\mathbf{q}$). □

In contrast, Proposition 2.11.1 is **false** for multigraphs. In fact, we can take a multigraph with a single vertex and lots of loops around it. In that case, its degree can be very large, but it has no cycles of length > 1.

3.3.9. $G \setminus e$, bridges and cut-edges

Next, we extend the definition of $G \setminus e$ (Definition 2.12.1) to multigraphs:

Definition 3.3.17. 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 this edge e. In other words,

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

$i < a < b \leq j$. Let us do this. Let a and b be two integers satisfying $i < a < b \leq j$. We must show that $e_a \neq e_b$. We distinguish two cases: the case $a = b - 1$ and the case $a \neq b - 1$.

- If $a = b - 1$, then e_a and e_b are two adjacent edges of $\mathbf{w}$ and thus distinct (since we assumed that no two adjacent edges of $\mathbf{w}$ are identical). Thus, $e_a \neq e_b$ is proved in the case when $a = b - 1$.
- Now, consider the case when $a \neq b - 1$. In this case, we must have $a < b - 1$ (since $a < b$ entails $a \leq b - 1$). Also, $i \leq a - 1$ (since $i < a$). Hence, $i \leq a - 1 < a < b - 1 \leq j - 1$ (since $b \leq j$). Therefore, $b - 1$, $a - 1$ and a are three distinct elements of the set $\{i, i+1, \ldots, j-1\}$. Consequently, w_{b-1}, w_{a-1}, w_a are three distinct vertices (since the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct). Therefore, $w_{b-1} \notin \{w_{a-1}, w_a\} = \varphi(e_a)$ (since $\mathbf{w}$ is a walk, so that the edge e_a has endpoints w_{a-1} and w_a). However, $\varphi(e_b) = \{w_{b-1}, w_b\}$ (since $\mathbf{w}$ is a walk, so that the edge e_b has endpoints w_{b-1} and w_b). Now, comparing $w_{b-1} \in \{w_{b-1}, w_b\} = \varphi(e_b)$ with $w_{b-1} \notin \varphi(e_a)$, we see that the sets $\varphi(e_b)$ and $\varphi(e_a)$ must be distinct (since $\varphi(e_b)$ contains w_{b-1} but $\varphi(e_a)$ does not). In other words, $\varphi(e_b) \neq \varphi(e_a)$. Hence, $e_b \neq e_a$. In other words, $e_a \neq e_b$. Thus, $e_a \neq e_b$ is proved in the case when $a \neq b - 1$.

We have now proved $e_a \neq e_b$ in both cases, so we are done.

Some authors write $G - e$ for $G \setminus e$.

The analogue of Theorem 2.12.2 for multigraphs holds (and can be proved in the same way as Theorem 2.12.2):

Theorem 3.3.18. 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 of vertices. It is these sets that we are talking about here, not the induced subgraphs on these sets.)

(b) If e appears in no cycle of G (in other words, there exists no cycle of G such that e is an edge of this cycle), then the graph $G \setminus e$ has one more component than G.

Note that an edge e that is a loop always is an edge of a cycle (indeed, it creates a cycle of length 1), and can never appear on any path; thus, removing such an edge e obviously does not change the path-connectedness relation.

Defining cut-edges and bridges just as we did for simple graphs (Definition 2.12.4), we equally recover the following corollary:

Corollary 3.3.19. Let e be an edge of a multigraph G. Then, e is a bridge if and only if e is a cut-edge.

Proof. Just like the proof of Corollary 2.12.5. $\square$

3.3.10. Dominating sets

We defined and studied dominating sets in Section 2.13. We could define dominating sets for multigraphs in the same way as for simple graphs, but we would not get anything new this way. Indeed, if G is a multigraph, then the dominating sets of G are precisely the dominating sets of G^{simp}. Thus, we can reduce any claims about dominating sets of multigraphs to analogous claims about simple graphs.

3.3.11. Hamiltonian paths and cycles

As we said before, a multigraph G has a Hamiltonian path or Hamiltonian cycle if and only if the corresponding simple graph G^{simp} has one. This does not mean, however, that everything we proved about Hamiltonian paths still applies to multigraphs. For instance, neither Ore's theorem (Theorem 2.14.4) nor Dirac's theorem (Corollary 2.14.5) holds for multigraphs, because we could duplicate edges to make degrees arbitrarily large, without necessarily creating a hamc.

Proposition 2.14.6 still holds for multigraphs, but this is clear because it can be derived from the corresponding property of G^{simp}.

3.3.12. Exercises

Exercise 3.1. Which of the Exercises 2.3, 2.4, 2.7, 2.14, 2.15, 2.5 and 2.8 remain true if "simple graph" is replaced by "multigraph"?

(For each exercise that becomes false, provide a counterexample. For each exercise that remains true, either provide a new solution that works for multigraphs, or argue that the solution we have seen applies verbatim to multigraphs, or derive the multigraph case from the simple graph case.)

Exercise 3.2. Let G be a multigraph with at least one edge. Assume that each vertex of G has even degree. Prove that G has a cycle.

[**Solution:** This is Exercise 4 on midterm #1 from my Spring 2017 course; see the course page for solutions.]

Exercise 3.3. Let G be a multigraph. Let $d > 2$ be an integer. Assume that $\deg v > 2$ for each vertex v of G. Prove that G has a cycle whose length is not divisible by d.

Exercise 3.4. Let G be a multigraph. Assume that G has exactly two vertices of odd degree. Prove that these two vertices are path-connected.

Exercise 3.5. Let $G = (V, E, \varphi)$ be a multigraph that has no loops.

If $e \in E$ is an edge that contains a vertex $v \in V$, then we let e/v denote the endpoint of e distinct from v. (If e is a loop, then this is understood to mean v itself.)

For each $v \in V$, we define a rational number q_v by

$$q_v = \sum_{\substack{e \in E; \\ v \in \varphi(e)}} \frac{\deg(e/v)}{\deg v}.$$

(Note that the denominator $\deg v$ on the right hand side is nonzero whenever the sum is nonempty!)

(Thus, q_v is the average degree of the neighbors of v, weighted with the number of edges that join v to the respective neighbors. If v has no neighbors, then $q_v = 0$.)

Prove that

$$\sum_{v \in V} q_v \geq \sum_{v \in V} \deg v.$$

(In other words, in a social network, your average friend has, on average, more friends than you do!)

[**Hint:** Any positive reals x and y satisfy $\dfrac{x}{y} + \dfrac{y}{x} \geq 2$. Why, and how does this help?]

Exercise 3.6. Let F be any field. (For instance, F can be $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$.)

Let $G = (V, E, \varphi)$ be a multigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

For each edge $e \in E$, we construct a column vector $\chi_e \in F^n$ (that is, a column vector with n entries) as follows:

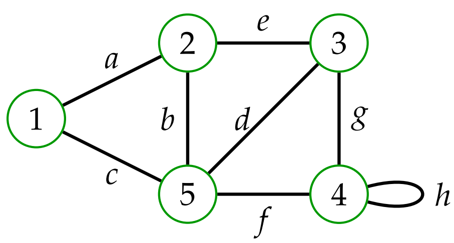
- If e is a loop, then we let χ_e be the zero vector.
- Otherwise, we let u and v be the two endpoints of e, and we let χ_e be the column vector that has a 1 in its u-th position, a -1 in its v-th position, and 0s in all other positions. (This depends on which endpoint we call u and which endpoint we call v, but we just make some choice and stick with it. The result will be true no matter how we choose.)

Let M be the $n \times |E|$-matrix over F whose columns are the column vectors χ_e for all $e \in E$ (we order them in some way; the exact order doesn't matter). Prove that

$$\operatorname{rank} M = |V| - \operatorname{conn} G,$$

where $\operatorname{conn} G$ denotes the number of components of G.

[**Example:** Here is an example: Let G be the multigraph



(so that $n = 5$). Then, if we choose the endpoints of b to be 2 and 5 in this order, then we have $\chi_b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$. (Choosing them to be 5 and 2 instead, we would obtain $\chi_b = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$.) If we do the same for all edges of G (that is, we choose the smaller endpoint as u and the larger endpoint as v), and if we order the columns so that they correspond to the edges a, b, c, d, e, f, g, h from left to right, then the matrix M comes out as follows:

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & -1 & 0 & -1 & 0 & 0 \end{pmatrix}.$$

It is easy to see that $\operatorname{rank} M = 4$, which is precisely $|V| - \operatorname{conn} G$.]

[**Remark:** The claim of the exercise can be restated as follows: The span of the vectors χ_e for all $e \in E$ has dimension $|V| - \operatorname{conn} G$.

Topologists will recognize the matrix M as (a matrix that represents) the boundary operator $\partial : C_1(G) \to C_0(G)$, where G is viewed as a CW-complex.]

Exercise 3.7. If G is a multigraph, then $\operatorname{conn} G$ shall denote the number of connected components of G. (Note that this is 0 when G has no vertices, and 1 if G is connected.)

Let (V, H, φ) be a multigraph. Let E and F be two subsets of H.

(a) Prove that

$$\operatorname{conn}(V,\ E,\ \varphi\mid_E) + \operatorname{conn}(V,\ F,\ \varphi\mid_F)$$
$$\leq \operatorname{conn}(V,\ E\cup F,\ \varphi\mid_{E\cup F}) + \operatorname{conn}(V,\ E\cap F,\ \varphi\mid_{E\cap F}). \qquad (2)$$

[**Hint:** Feel free to restrict yourself to the case of a simple graph; in this case, E and F are two subsets of $\mathcal{P}_2(V)$, and you have to show that

$$\operatorname{conn}(V,\ E) + \operatorname{conn}(V,\ F) \leq \operatorname{conn}(V,\ E\cup F) + \operatorname{conn}(V,\ E\cap F).$$

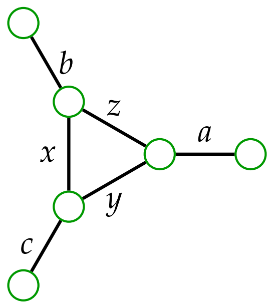
This isn't any easier than the general case, but saves you the hassle of carrying the map φ around.]

(b) Give an example where the inequality (2) does **not** become an equality.

[**Solution:** This is Exercise 3 on homework set #3 from my Spring 2017 course; see the course page for solutions.]

Exercise 3.8. Let $G = (V, E, \varphi)$ be a connected multigraph with $2m$ edges, where $m \in \mathbb{N}$. A set $\{e, f\}$ of two distinct edges will be called a **friendly couple** if e and f have at least one endpoint in common. Prove that the edge set of G can be decomposed into m disjoint friendly couples (i.e., there exist m disjoint friendly couples $\{e_1, f_1\}, \{e_2, f_2\}, \ldots, \{e_m, f_m\}$ such that $E = \{e_1, f_1, e_2, f_2, \ldots, e_m, f_m\}$). ("Disjoint" means "disjoint as sets" – i.e., having no edges in common.)

[**Example:** Here is a graph with an even number of edges:



One possible decomposition of its edge set into disjoint friendly couples is $\{a, y\}, \{b, z\}, \{c, x\}$.]

[**Hint:** Induct on $|E|$. Pick a vertex v of degree > 1 and consider the components of $G \setminus v$.]

Exercise 3.9. Let $n \geq 0$. Let $d_1, d_2, \ldots, d_n$ be n nonnegative integers such that $d_1 + d_2 + \cdots + d_n$ is even.

(a) Prove that there exists a multigraph G with vertex set $\{1, 2, \ldots, n\}$ such that all $i \in \{1, 2, \ldots, n\}$ satisfy $\deg i = d_i$.

(b) Prove that there exists a loopless multigraph G with vertex set $\{1, 2, \ldots, n\}$ such that all $i \in \{1, 2, \ldots, n\}$ satisfy $\deg i = d_i$ if and only if each $i \in \{1, 2, \ldots, n\}$ satisfies the inequality

$$\sum_{\substack{j \in \{1,2,\ldots,n\}; \\ j \neq i}} d_j \geq d_i. \tag{3}$$

[**Remark:** The inequality (3) is the "n-gon inequality": It is equivalent to the existence of a (possibly degenerate) n-gon with sidelengths $d_1, d_2, \ldots, d_n$.]

Exercise 3.10. Let G be a loopless multigraph. Recall that a **trail** (in G) means a walk whose edges are distinct (but whose vertices are not necessarily distinct). Let u and v be two vertices of G. As usual, "trail from u to v" means "trail that starts at u and ends at v". Prove that

$$\begin{aligned} &(\text{the number of trails from } u \text{ to } v \text{ in } G) \\ &\equiv (\text{the number of paths from } u \text{ to } v \text{ in } G) \bmod 2. \end{aligned}$$

[**Hint:** Try to pair up the non-path trails into pairs. Make sure to prove that this pairing is well-defined (i.e., each non-path trail $\mathbf{t}$ has exactly one partner, which is not itself, and that $\mathbf{t}$ is the designated partner of its partner!).]

Exercise 3.11. Let G be a multigraph such that every vertex of G has even degree. Let u and v be two distinct vertices of G. Prove that the number of paths from u to v is even.

[**Hint:** When you add an edge joining u to v, the graph G becomes a graph with exactly two odd-degree vertices u and v, and the claim becomes "the number of paths from u to v is odd" (why?). In this form, the claim turns out to be easier to prove. Indeed, any path must start with some edge...
Keep in mind that paths can be replaced by trails, by Exercise 3.10.]

Exercise 3.12. Let $G = (V, E, \varphi)$ be a multigraph such that $|E| > |V|$. Prove that G has a cycle of length $\leq \dfrac{2n+2}{3}$, where $n = |V|$.

[**Solution:** This is Exercise 8 on midterm #3 from my Spring 2017 course (except that the simple graph was replaced by a multigraph); see the course page for solutions.]

3.4. Eulerian circuits and walks

3.4.1. Definitions

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

Recall that a Hamiltonian path or cycle is a path or cycle that contains all vertices of the graph. Being a path or cycle, it has to contain each of them exactly once (except, in the case of a cycle, of its starting point).

What about a walk or closed walk that contains all **edges** exactly once instead? These are called "Eulerian" walks or circuits; here is the formal definition:

Definition 3.4.1. Let G be a multigraph.

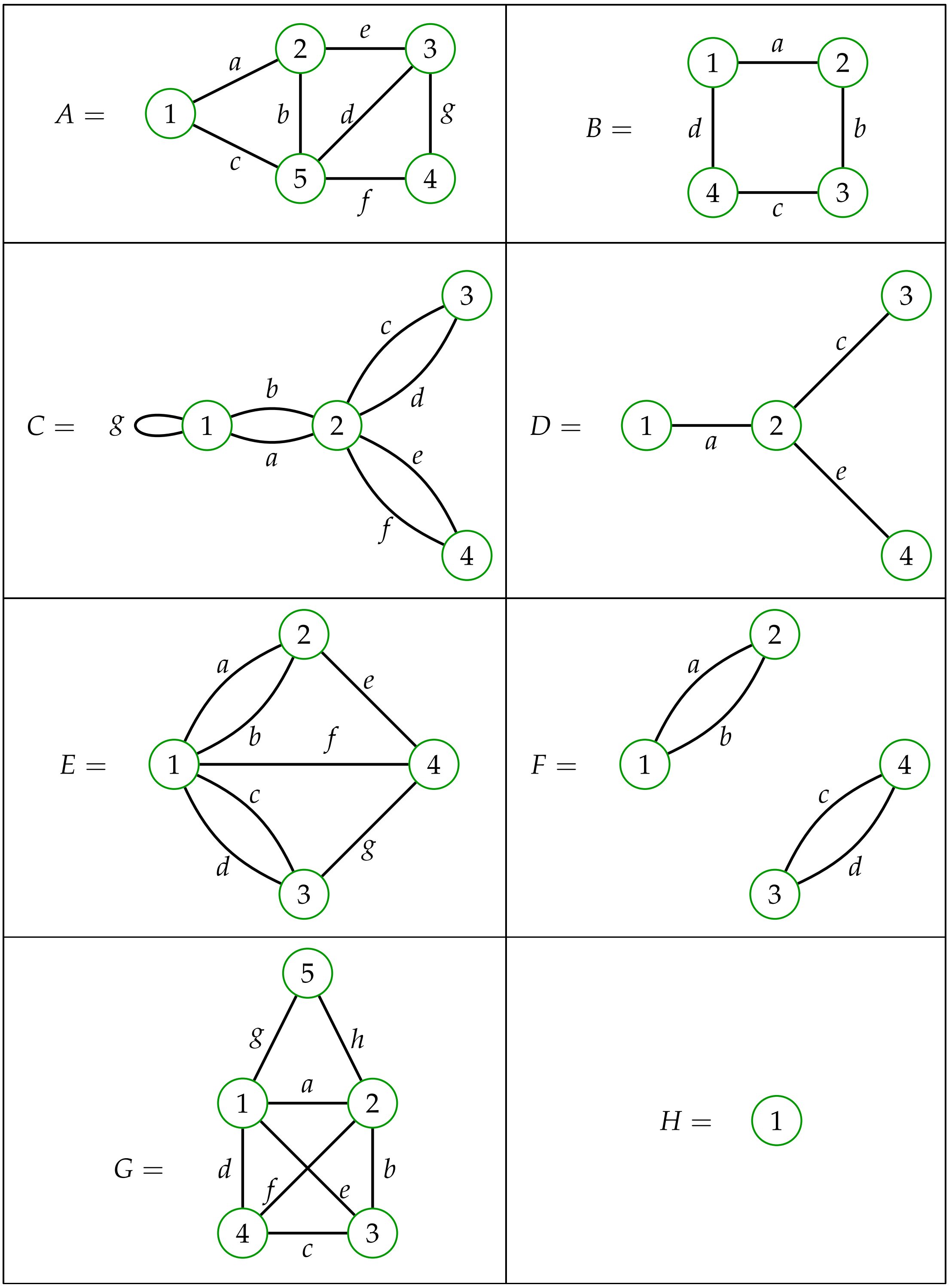
(a) A walk of G is said to be **Eulerian** if each edge of G appears exactly once in this walk.

(In other words: A walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ of G is said to be **Eulerian** if for each edge e of G, there exists exactly one $i \in \{1, 2, \ldots, k\}$ such that $e = e_i$.)

(b) An **Eulerian circuit** of G means a circuit (i.e., closed walk) of G that is Eulerian. (Strictly speaking, the preceding sentence is redundant, but we still said it to stress the notion of an Eulerian circuit.)

Unlike for Hamiltonian paths and cycles, an Eulerian walk or circuit is usually not a path or cycle. Also, finding an Eulerian walk in a multigraph G is not the same as finding an Eulerian walk in the simple graph G^{simp}. (Nevertheless, some authors call Eulerian walks "Eulerian paths" and call Eulerian circuits "Eulerian cycles". This is rather confusing.)

Example 3.4.2. Consider the following multigraphs:

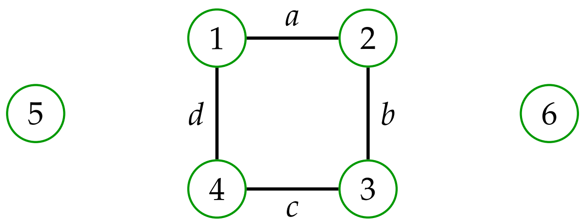


- The multigraph A has an Eulerian walk

$(3, d, 5, b, 2, e, 3, g, 4, f, 5, c, 1, a, 2)$. But A has no Eulerian circuit. The easiest way to see this is by observing that A has a vertex of odd degree (e.g., the vertex 2). If an Eulerian circuit were to exist, then it would have to enter this vertex as often as it exited it; but this would mean that the degree of this vertex would be even (because each edge containing this vertex would be used exactly once either to enter or to exit it, except for loops, which would be used twice). So, more generally, any multigraph that has a vertex of odd degree cannot have an Eulerian circuit.

- The multigraph B has an Eulerian circuit $(1, a, 2, b, 3, c, 4, d, 1)$, and thus of course an Eulerian walk (since any Eulerian circuit is an Eulerian walk).
- The multigraph C has an Eulerian circuit $(1, g, 1, b, 2, c, 3, d, 2, e, 4, f, 2, a, 1)$.
- The multigraph D has no Eulerian walk. Indeed, it has four vertices of odd degree. If v is a vertex of odd degree, then any Eulerian walk has to either start or end at v (since otherwise, the walk would enter and leave v equally often, but then the degree of v would be even). But a walk can only have one starting point and one ending point. This allows for two vertices of odd degree, but not more than two. So, more generally, any multigraph that has more than two vertices of odd degree cannot have an Eulerian walk.
- The multigraph E has no Eulerian walk. The reason is the same as for D. Note that E is the famous multigraph of bridges in Königsberg, as studied by Euler in 1736 (see the Wikipedia page for "Seven bridges of Königsberg" for the backstory).
- The multigraph F has no Eulerian walk, since it has two components, each containing at least one edge. (An Eulerian walk would have to contain both edges b and c, but there is no way to walk between them, since they belong to different components.)
- The multigraph G has an Eulerian walk, namely $(3, b, 2, h, 5, g, 1, a, 2, f, 4, d, 1, e, 3, c, 4)$. It has no Eulerian circuit, since it has two vertices of odd degree.
- The multigraph H has an Eulerian circuit, namely (1).

Remark 3.4.3. For the pedants: A multigraph can have an Eulerian circuit even if it is not connected, as long as all its edges belong to the same component (i.e., all but one components are just singletons with no edges). Here is an example:



Exercise 3.13. Let n be a positive integer. Recall from Definition 2.6.1 **(a)** that K_n denotes the complete graph on n vertices. This is the graph with vertex set $V = \{1, 2, \ldots, n\}$ and edge set $\mathcal{P}_2(V)$ (so each two distinct vertices are adjacent).

Find Eulerian circuits for the graphs K_3, K_5, and K_7.

[**Solution:** This is Exercise 2 on homework set #2 from my Spring 2017 course; see the course page for solutions.]

3.4.2. The Euler–Hierholzer theorem

How hard is it to find an Eulerian walk or circuit in a multigraph, or to check if there is any? Surprisingly, this is a lot easier than the same questions for Hamiltonian paths or cycles. The second question in particular is answered (for connected multigraphs) by the **Euler–Hierholzer theorem**:

Theorem 3.4.4 (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.

We already proved the "$\Longrightarrow$" directions of both parts **(a)** and **(b)** in Example 3.4.2. It remains to prove the "$\Longleftarrow$" directions. I don't think that Euler actually proved them in his 1736 paper, but Hierholzer did in 1873. The "standard" proof can be found in many texts, such as [Guicha16, Theorem 5.2.2 and Theorem 5.2.3]. I will sketch a different proof, which I learnt from [LeLeMe18, Problem 12.35]. We begin with the following definition:

Definition 3.4.5. 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 cannot repeat edges.

Thus, an Eulerian walk has to be a trail. A trail cannot be any longer than an Eulerian walk. So a reasonable way to try constructing an Eulerian walk is to start with some trail, and make it progressively longer until it becomes Eulerian (hopefully).

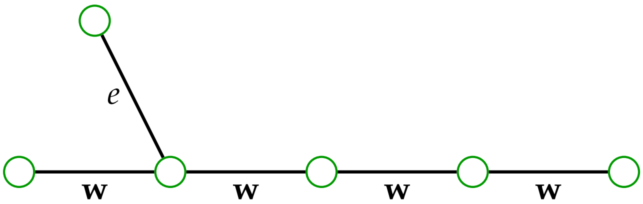
This suggests the following approach to proving the "$\Longleftarrow$" directions of Theorem 3.4.4: We pick the longest trail of G and argue that (under the right assumptions) it has to be Eulerian, since otherwise there would be a way to make it longer. Of course, we need to find such a way. Here is the first step:

Lemma 3.4.6. Let G be a multigraph with at least one vertex. Then, G has a longest trail.

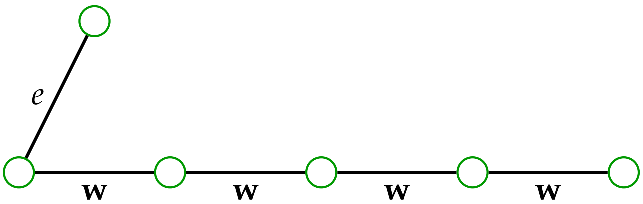
Proof. Clearly, G has at least one trail (e.g., a length-0 trail from a vertex to itself). Moreover, G has only finitely many trails (since each edge of G can only be used once in a trail, and there are only finitely many edges). Hence, the maximum principle proves the lemma. $\square$

Our goal now is to show that under appropriate conditions, such a longest trail will be Eulerian. This will require two further lemmas.

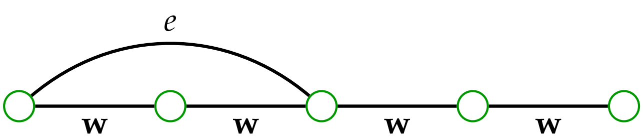
First, one more piece of notation: We say that an edge e of a multigraph G **intersects** a walk $\mathbf{w}$ if at least one endpoint of e is a vertex of $\mathbf{w}$. Here is how this can look like:



(here, the edges of $\mathbf{w}$ are marked with a "$\mathbf{w}$" underneath them) or



(here, the endpoint of e that is a vertex of $\mathbf{w}$ happens to be the starting point of $\mathbf{w}$) or



(here, both endpoints of e happen to be vertices of $\mathbf{w}$). Be careful with such pictures, though: A walk doesn't have to be a path; it can visit a vertex any number of times!

Lemma 3.4.7. 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. We assumed that there exists an edge of G that is not an edge of $\mathbf{w}$. Pick such an edge, and call it f.

A "$\mathbf{w}$-f-path" will mean a path from a vertex of $\mathbf{w}$ to an endpoint of f. Such a path clearly exists, since G is connected. Thus, we can pick a **shortest** such path. If this shortest path has length 0, then we are done (since f intersects $\mathbf{w}$ in this case). If not, we consider the first edge of this path. This first edge cannot be an edge of $\mathbf{w}$, because otherwise we could remove it from the path and get an even shorter $\mathbf{w}$-f-path. But it clearly intersects $\mathbf{w}$. So we have found an edge of G that is not an edge of $\mathbf{w}$ but intersects $\mathbf{w}$. This proves the lemma. □

Lemma 3.4.8. 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 a closed walk, we thus have $u \neq v$.

Consider the edges of $\mathbf{w}$ that contain v. Such edges are of two kinds: those by which $\mathbf{w}$ enters v (this means that v comes immediately after this edge in $\mathbf{w}$), and those by which $\mathbf{w}$ leaves v (this means that v comes immediately before this edge in $\mathbf{w}$). [18] 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; conversely, each edge of the latter kind is immediately preceded by an edge of the former kind (since $\mathbf{w}$ starts at the vertex u, which is distinct from v). Hence, the walk $\mathbf{w}$ has exactly one more edge entering v than it has edges leaving v. Thus, the number of edges of $\mathbf{w}$ that contain v (with loops counting twice) is odd. However, the total number of edges of G that contain v (with loops counting twice) is even (because it is the degree of v, but we assumed that each vertex of G has even degree). So these two numbers are distinct. Thus, there is at least one edge of G that contains v but is not an edge of $\mathbf{w}$.

Fix such an edge and call it f. Now, append f to the trail $\mathbf{w}$ at the end. The result will be a trail (since f is not an edge of $\mathbf{w}$) that is longer than $\mathbf{w}$. But this contradicts the fact that $\mathbf{w}$ is a longest trail. Thus, the lemma is proved. □

We can now finish the proof of the Euler–Hierholzer theorem:

Proof of Theorem 3.4.4. **(a)** $\Longrightarrow$: We proved this back in Example 3.4.2.

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

By Lemma 3.4.6, we know that G has a longest trail. Fix such a longest trail, and call it $\mathbf{w}$. Then, Lemma 3.4.8 shows that $\mathbf{w}$ is a closed walk.

[18] Loops whose only endpoint is v count as both.

We claim that $\mathbf{w}$ is Eulerian. Indeed, assume the contrary. Then, there exists an edge of G that is not an edge of $\mathbf{w}$. Hence, Lemma 3.4.7 shows that there exists an edge of G that is not an edge of $\mathbf{w}$ but intersects $\mathbf{w}$. Fix such an edge, and call it f.

Since f intersects $\mathbf{w}$, there exists an endpoint v of f that is a vertex of $\mathbf{w}$. Consider this v. Since $\mathbf{w}$ is a **closed** trail, we can WLOG assume that $\mathbf{w}$ starts and ends at v (since we can otherwise achieve this by rotating[19] $\mathbf{w}$). Then, we can append the edge f to the trail $\mathbf{w}$. This results in a new trail (since f is not an edge of $\mathbf{w}$) that is longer than $\mathbf{w}$. And this contradicts the fact that $\mathbf{w}$ is a longest trail of G.

This contradiction proves that $\mathbf{w}$ is Eulerian. Hence, $\mathbf{w}$ is an Eulerian circuit (since $\mathbf{w}$ is a closed walk). Thus, the "$\Longleftarrow$" direction of Theorem 3.4.4 **(a)** is proven.

(b) $\Longrightarrow$: Already proved in Example 3.4.2.

$\Longleftarrow$: Assume that all but at most two vertices of G have even degree. We must prove that G has an Eulerian walk.

If each vertex of G has even degree, then this follows from Theorem 3.4.4 **(a)**, since every Eulerian circuit is an Eulerian walk. Thus, we WLOG assume that not each vertex of G has even degree. In other words, the number of vertices of G having odd degree is positive.

The handshake lemma for multigraphs (i.e., Corollary 3.3.3) shows that the number of vertices of G having odd degree is even. Furthermore, this number is at most 2 (since all but at most two vertices of G have even degree). So this number is even, positive and at most 2. Thus, this number is 2. In other words, the multigraph G has exactly two vertices having odd degree. Let u and v be these two vertices.

Add a new edge e that has endpoints u and v to the multigraph G (do this even if there already is such an edge![20]). Let G' denote the resulting multigraph. Then, in G', each vertex has even degree (since the newly added edge e has increased the degrees of u and v by 1, thus turning them from odd to even). Moreover, G' is still connected (since G was connected, and the newly added edge e can hardly take that away). Thus, we can apply Theorem 3.4.4 **(a)** to G' instead of G. As a result, we conclude that G' has an Eulerian circuit. Cutting the newly added edge e out of this Eulerian circuit[21], we obtain an Eu-

[19] **Rotating** a closed walk $(w_0, e_1, w_1, e_2, w_2, \ldots, e_k, w_k)$ means moving its first vertex and its first edge to the end, i.e., replacing the walk by $(w_1, e_2, w_2, e_3, w_3, \ldots, e_k, w_k, e_1, w_1)$. This always results in a closed walk again. For example, if $(1, a, 2, b, 3, c, 1)$ is a closed walk, then we can rotate it to obtain $(2, b, 3, c, 1, a, 2)$; then, rotating it one more time, we obtain $(3, c, 1, a, 2, b, 3)$.

Clearly, by rotating a closed walk several times, we can make it start at any of its vertices. Moreover, if we rotate a closed trail, then we obtain a closed trail.

[20] This is a time to be grateful for the notion of a multigraph. We could not do this with simple graphs!

[21] More precisely: We rotate this circuit until e becomes its last edge, and then we remove this last edge to obtain a walk.

lerian walk of G. Hence, G has an Eulerian walk. Thus, the "$\Longleftarrow$" direction of Theorem 3.4.4 **(b)** is proven. $\square$

Note: If you look closely at the above proof, you will see hidden in it an algorithm for **finding** Eulerian circuits and walks.[22]

Exercise 3.14. Let G be a connected multigraph. Let m be the number of vertices of G that have odd degree. Prove that we can add $m/2$ new edges to G in such a way that the resulting multigraph will have an Eulerian circuit. (It is allowed to add an edge even if there is already an edge between the same two vertices.)

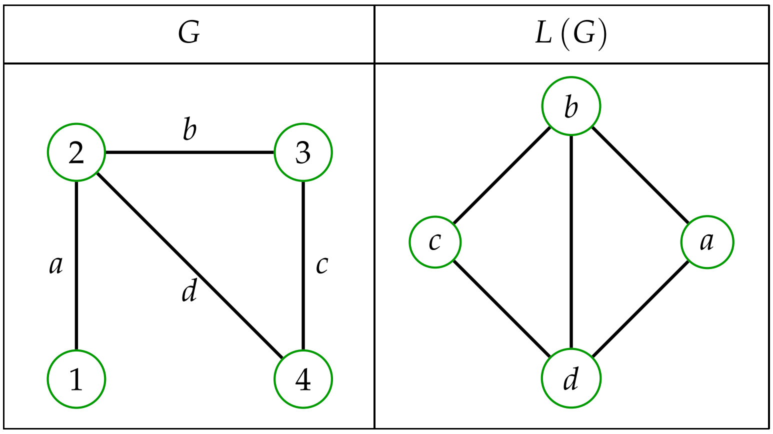
[**Solution:** This exercise is Exercise 6 on midterm #1 from my Spring 2017 course; see the course page for solutions.]

Exercise 3.15. Let $G = (V, E, \varphi)$ be a multigraph. The **line graph** $L(G)$ is defined as the simple graph (E, F), where

$$F = \{\{e_1, e_2\} \in \mathcal{P}_2(E) \mid \varphi(e_1) \cap \varphi(e_2) \neq \varnothing\}.$$

(In other words, $L(G)$ is the graph whose **vertices** are the **edges** of G, and in which two vertices e_1 and e_2 are adjacent if and only if the edges e_1 and e_2 of G share a common endpoint.)

[**Example:** Here is a multigraph G along with its line graph $L(G)$:



.

Note that $L(G)$ does not always determine G uniquely.]

Assume that $|V| > 1$. Prove the following:

[22] You might be skeptical about this. After all, in order to apply Lemma 3.4.8, we need a longest trail, so you might wonder how we can find a longest trail to begin with.

Fortunately, we don't need to take Lemma 3.4.8 this literally. Our above proof of Lemma 3.4.8 can be used even if $\mathbf{w}$ is **not** a longest trail. In this case, however, instead of showing that $\mathbf{w}$ is a closed walk, this proof may show us a way how to make $\mathbf{w}$ longer. In other words, by following this proof, we may discover a trail longer than $\mathbf{w}$. In this case, we can replace $\mathbf{w}$ by this longer trail, and then apply Lemma 3.4.8 again. We can repeat this over and over again, until we do end up with a closed walk. (This will eventually happen, since we know that a trail cannot be longer than the total number of edges of G.)

(a) If G has a Hamiltonian path, then $L(G)$ has a Hamiltonian path.

(b) If G has an Eulerian walk, then $L(G)$ has a Hamiltonian path.

[**Solution:** This exercise is Exercise 2 on midterm #1 from my Spring 2017 course (generalized from simple graphs to multigraphs); see the course page for solutions.]

4. Digraphs and multidigraphs

4.1. Definitions

We have so far seen two concepts of graphs: simple graphs and multigraphs.

For all their differences, these two concepts have one thing in common: The two endpoints of an edge are equal in rights. Thus, when defining walks, each edge serves as a "two-way road". Hence, such graphs are good at modelling symmetric relations between things.

We shall now introduce two analogous versions of "graphs" in which the edges have directions. These versions 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 will draw these edges as arrows, and we will only allow using them in one direction (viz., from source to target) when we walk down the graph. Here are the definitions in detail:

Definition 4.1.1. 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 4.1.2. 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 **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.

When u and v are two elements of V, we will occasionally use uv as a shorthand for the pair (u, v). Note that this means an ordered pair now!

(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.

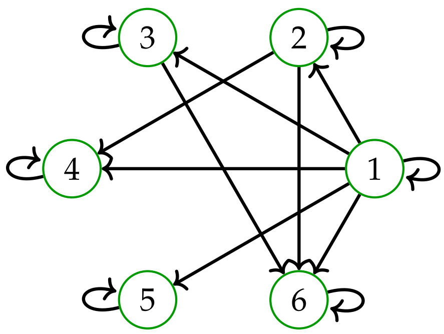
(d) We draw D as follows: We represent each vertex of D by a point, and each arc uv by an arrow that goes from the point representing u to the point representing v.

(e) An arc (u, v) is called a **loop** (or **self-loop**) if $u = v$. (In other words, an arc is a loop if and only if its source is its target.)

Example 4.1.3. For each $n \in \mathbb{N}$, we define the **divisibility digraph on** $\{1, 2, \ldots, n\}$ to be the simple digraph (V, A), where $V = \{1, 2, \ldots, n\}$ and

$$A = \{(i, j) \in V \times V \mid i \text{ divides } j\}.$$

For example, for $n = 6$, this digraph looks as follows:



. (4)

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

Definition 4.1.4. 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 4.1.5. 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 **nodes**) of D.

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

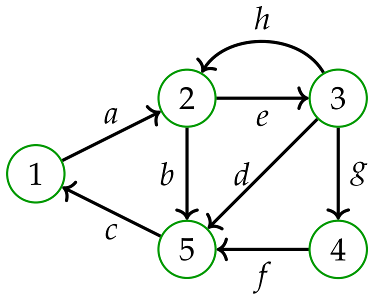
Its elements are called the **arcs** (or **directed edges**) of D.

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

(d) We draw D as follows: We represent each vertex of D by a point, and each arc a by an arrow that goes from the point representing u to the point representing v, where $(u, v) = \psi(a)$.

Example 4.1.6. Here is a multidigraph:

$D =$



. (5)

Formally speaking, this multidigraph is the triple (V, A, ψ), where $V = \{1, 2, 3, 4, 5\}$ and $A = \{a, b, c, d, e, f, g, h\}$ and $\psi(a) = (1, 2)$ and $\psi(b) = (2, 5)$ and so on.

Thus, simple digraphs and multidigraphs are analogues of simple graphs and multigraphs, respectively, in which the edges have been replaced by arcs ("edges endowed with a direction"). The analogy is perfect but for the fact that simple graphs forbid loops but simple digraphs allow loops (but different authors have different opinions on this).

Convention 4.1.7. The word "**digraph**" means either "simple digraph" or "multidigraph", depending on the context.

The word "digraph" was originally a shorthand for "**directed graph**", but by now it is a technical term that is perfectly understood by everyone in the subject. (It is also understood by linguists, but in a rather different way.)

4.2. Outdegrees and indegrees

What can we do with digraphs? Many of the things we have done with graphs can be modified to work with digraphs (although not all their properties will still hold). For example, the notion of the degree of a vertex in a graph has the following two counterpart notions for digraphs:

Definition 4.2.1. Let D be a digraph with vertex set V. (This can be either a simple digraph or a multidigraph.) 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 $\deg^+ v$.

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

Example 4.2.2. In the divisibility digraph on $\{1,2,3,4,5,6\}$ (see (4) for a drawing), we have

$$\begin{array}{llll} \deg^+ 1 = 6, & \deg^- 1 = 1, & \deg^+ 2 = 3, & \deg^- 2 = 2, \\ \deg^+ 3 = 2, & \deg^- 3 = 2, & \deg^+ 4 = 1, & \deg^- 4 = 3, \\ \deg^+ 5 = 1, & \deg^- 5 = 2, & \deg^+ 6 = 1, & \deg^- 6 = 4. \end{array}$$

Recall Euler's result (Proposition 3.3.2) saying that in a graph, the sum of all degrees is twice the number of edges. Here is an analogue of this result for digraphs:

Proposition 4.2.3 (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 an outdegree, we have

$$\deg^+ v = (\text{the number of arcs of } D \text{ whose source is } v)$$

for each $v \in V$. Thus,

$$\begin{aligned} \sum_{v\in V} \deg^+ v &= \sum_{v\in V} (\text{the number of arcs of } D \text{ whose source is } v) \\ &= (\text{the number of all arcs of } D) \\ &\qquad \left(\begin{array}{c}\text{since each arc of } D \text{ has exactly one source,} \\ \text{and thus is counted exactly once in the sum}\end{array}\right) \\ &= |A|. \end{aligned}$$

Similarly, $\sum_{v\in V} \deg^- v = |A|$. ☐

("diEuler" is not a real mathematician; I just gave that moniker to Proposition 4.2.3 in order to stress its analogy with Euler's 1736 result.)

4.3. Subdigraphs

Just as we defined subgraphs of a multigraph, we can define subdigraphs (or "submultidigraphs", to be very precise) of a digraph:

Definition 4.3.1. Let $D = (V, A, \psi)$ be a multidigraph.

(a) A **submultidigraph** (or, for short, **subdigraph**) of D means a multidigraph of the form $E = (W, B, \chi)$, where $W \subseteq V$ and $B \subseteq A$ and

$\chi = \psi\mid_B$. In other words, a submultidigraph of D means a multidigraph E whose vertices are vertices of D and whose arcs are arcs of D and whose arcs have the same sources and targets in E as they have in D.

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

$$\left(S,\ \ A',\ \ \psi\mid_{A'}\right)$$

of D, where

$$A' := \{a \in A \ \mid \text{ both the source and the target of } a \text{ belong to } S\}.$$

In other words, it denotes the subdigraph of D whose vertices are the elements of S, and whose arcs are precisely those arcs of D whose sources and targets both belong to S. We denote this induced subdigraph by $D[S]$.

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

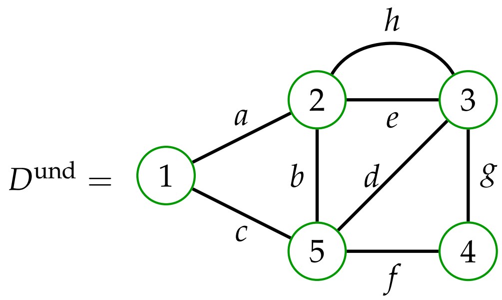
4.4. Conversions

4.4.1. 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 4.4.1. Let D be a multidigraph. 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, this is defined as follows: If $D = (V, A, \psi)$, then $D^{\text{und}} = (V, A, \varphi)$, where the map $\varphi : A \to \mathcal{P}_{1,2}(V)$ sends each arc $a \in A$ to the set of the entries of $\psi(a)$ (that is, to the set consisting of the source of a and the target of a).

For example, if D is the multidigraph from (5), then D^{und} is the following multigraph:



.

4.4.2. Multigraphs to multidigraphs

We have just seen how to turn any multidigraph D into a multigraph D^{und} by forgetting the directions of the arcs.

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). Here is a formal definition:

Definition 4.4.2. 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 then be called t_e. (If e is a loop, then we understand t_e to mean s_e.)

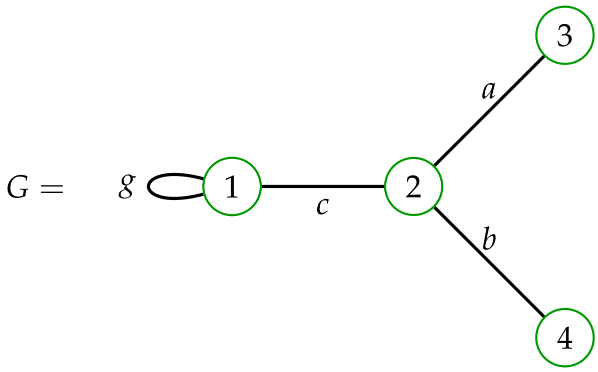
We then define G^{bidir} to be the multidigraph $(V,\ E \times \{1,2\}\,,\ \psi)$, where the map $\psi : E \times \{1,2\} \to V \times V$ is defined as follows: For each edge $e \in E$, we set

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

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

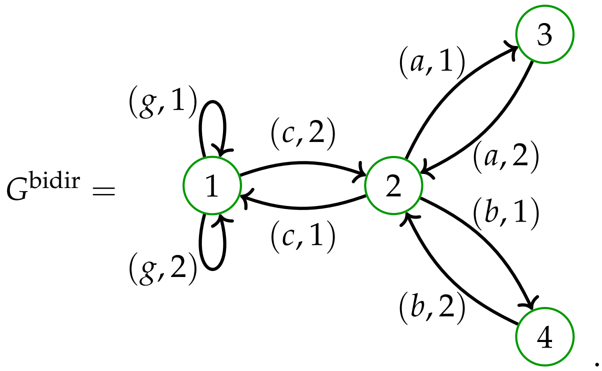
Note that the map ψ depends on our choice of s_e's (that is, it depends on which endpoint of an edge e we choose to be s_e). This makes the definition of G^{bidir} non-canonical; I don't know if there is a good way to fix this. Fortunately, all choices of s_e's will lead to mutually isomorphic multidigraphs G^{bidir}. (The notion of **isomorphism** for multidigraphs is exactly the one that you expect.)

Example 4.4.3. If



,

then



(Here, for example, we have chosen s_a to be 2, so that $t_a = 3$ and $\psi(a,1) = (2,3)$ and $\psi(a,2) = (3,2)$.) Yes, even the loops of G are duplicated in G^{bidir} !

The operation that assigns a multidigraph G^{bidir} to a multigraph G is injective – i.e., the original graph G can be uniquely reconstructed from G^{bidir}. This is in stark difference to the operation $D \mapsto D^{\mathrm{und}}$, which destroys information (the directions of the arcs). Note that the multigraph $\left(G^{\mathrm{bidir}}\right)^{\mathrm{und}}$ is not isomorphic to G, since each edge of G is doubled in $\left(G^{\mathrm{bidir}}\right)^{\mathrm{und}}$.

4.4.3. Simple digraphs to multidigraphs

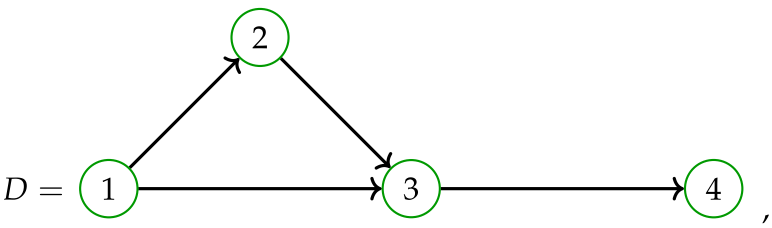
Next, we introduce another operation: one that turns simple digraphs into multidigraphs. This is very similar to the operation $G \mapsto G^{\mathrm{mult}}$ that turns simple graphs into multigraphs, so we will even use the same notation for it. Its definition is as follows:

Definition 4.4.4. Let $D = (V, A)$ be a simple digraph. Then, the **corresponding multidigraph** D^{mult} is defined to be the multidigraph

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

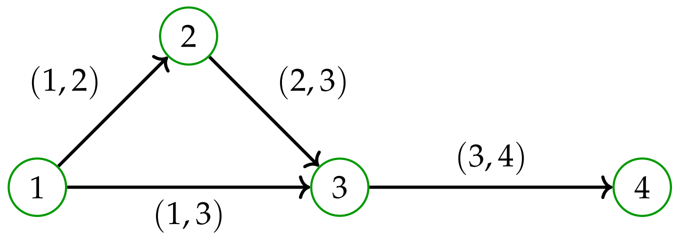
where $\iota : A \to V \times V$ is the map sending each $a \in A$ to a itself.

Example 4.4.5. If



,

then

$D^{\text{mult}} =$ 

.

4.4.4. Multidigraphs to simple digraphs

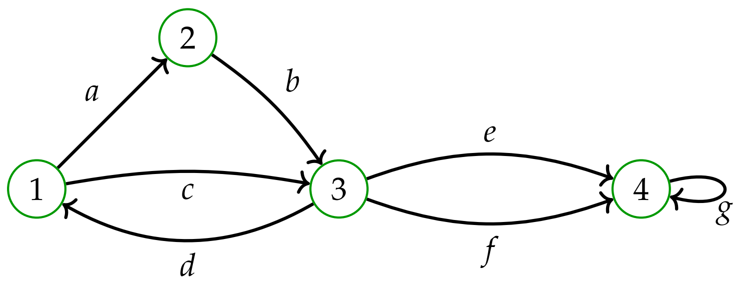
There is also an operation $D \mapsto D^{\text{simp}}$ that turns multidigraphs into simple digraphs:[23]

Definition 4.4.6. Let $D = (V, A, \psi)$ be a multidigraph. Then, the **underlying simple digraph** D^{simp} of D means the simple digraph

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

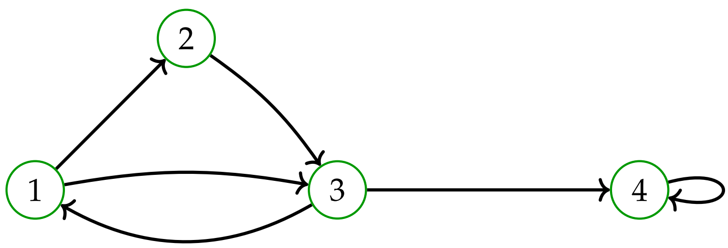
In other words, it is the simple digraph with vertex set V in which there is an arc from u to v if there exists an arc from u to v in D. Thus, D^{simp} is obtained from D by "collapsing" parallel arcs (i.e., arcs having the same source and the same target) to a single arc.

Example 4.4.7. If

$D =$ 

,

then

$D^{\text{simp}} =$ 

.

[23] I will use a notation that I probably should have introduced before: If u and v are two vertices of a digraph, then an "**arc from** u **to** v" means an arc with source u and target v.

Note that the arcs c and d have not been "collapsed" into one arc, since they do not have the same source and the same target. Likewise, the loop g has been preserved (unlike for undirected graphs).

4.4.5. Multidigraphs as a big tent

A takeaway from this all is that multidigraphs are the "most general" notion of graphs we have introduced so far. Indeed, using the operations we have seen so far, we can convert every notion of graphs into a multidigraph:

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

Since all three of these operations are injective (i.e., lose no information), we thus can encode each of our four notions of graphs as a multidigraph. Consequently, any theorem about multidigraphs can be specialized to the other three types of graphs. This doesn't mean that any theorem on any other type of graphs can be generalized to multidigraphs, though (e.g., Mantel's theorem holds only for simple graphs) – but when it can, we will try to state it at the most general level possible, to avoid doing the same work twice.

4.5. Walks, paths, closed walks, cycles

4.5.1. Definitions

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

For simple digraphs, we imitate the definitions from Sections 2.9 and 2.10 as best as we can, making sure to require all arcs to be traversed in the correct direction:

Definition 4.5.1. 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, v_2v_3, $\ldots$, $v_{k-1}v_k$ 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:

- The **vertices** of $\mathbf{w}$ are defined to be $v_0, v_1, \ldots, v_k$.

- The **arcs** of **w** are defined to be the pairs v_0v_1, v_1v_2, v_2v_3, $\ldots$, $v_{k-1}v_k$.
- The nonnegative integer k is called the **length** of **w**. (This is the number of all arcs of **w**, counted with multiplicity. It is 1 smaller than the number of all vertices of **w**, counted with multiplicity.)
- The vertex v_0 is called the **starting point** of **w**. We say that **w starts** (or **begins**) at v_0.
- The vertex v_k is called the **ending point** of **w**. We say that **w ends** at v_k.

(c) A **path** (in D) means a walk (in D) whose vertices are distinct. In other words, a path means a walk $(v_0, v_1, \ldots, v_k)$ such that $v_0, v_1, \ldots, v_k$ are distinct.

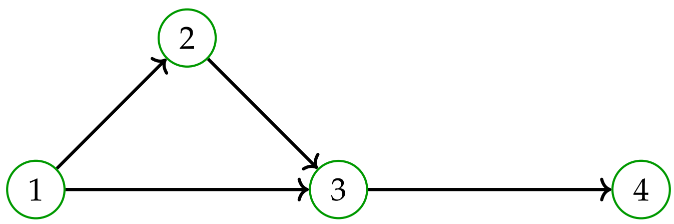
(d) Let p and q be two vertices of D. A **walk from** p **to** q means a walk that starts at p and ends at q. A **path from** p **to** q means a path that starts at p and ends at q.

(e) A **closed walk** of D 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 $w_0 = w_k$. Sometimes, closed walks are also known as **circuits** (but many authors use this latter word for something slightly different).

(f) A **cycle** of D means a closed walk $(w_0, w_1, \ldots, w_k)$ such that $k \geq 1$ and such that the vertices $w_0, w_1, \ldots, w_{k-1}$ are distinct.

Note that we replaced the condition $k \geq 3$ by $k \geq 1$ in the definition of a cycle, since simple digraphs can have loops. Fortunately, with the arcs being directed, we no longer have to worry about the same arc being traversed back and forth, so we need no extra condition to rule this out.

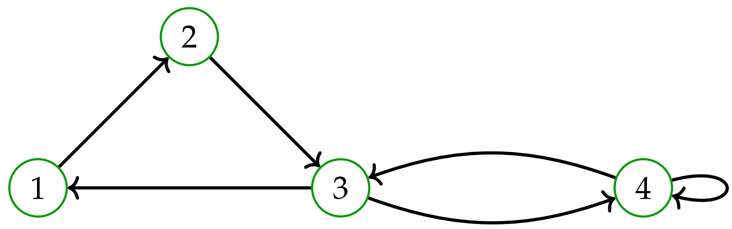
Example 4.5.2. Consider the simple digraph

$D =$ 

.

Then, $(1, 2, 3, 4)$ and $(1, 3, 4)$ are two walks of D, and these walks are paths. But $(2, 3, 1)$ is not a walk (since you cannot use the arc 13 to get from 3 to 1). This digraph D has no cycles, and its only closed walks have length 0.

Example 4.5.3. Consider the simple digraph

$D =$  .

Then, $(1, 2, 3, 1)$ and $(3, 4, 3)$ and $(4, 4)$ are cycles of D. Moreover, $(1, 2, 3, 4, 3, 1)$ is a closed walk but not a cycle.

Now let's define the same concepts for multidigraphs, by modifying the analogous definitions for multigraphs we saw in Definition 3.1.4:

Definition 4.5.4. 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, where $a_1, a_2, \ldots, a_k$ are arcs of D, and where each $i \in \{1, 2, \ldots, k\}$ satisfies

$$\psi(a_i) = (v_{i-1}, v_i)$$

(that is, each arc a_i has source v_{i-1} and target v_i). Note that we have to record both the vertices **and** the arcs in our walk, since we want the walk to "know" which arcs it traverses.

The **vertices** of a walk $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ are $v_0, v_1, \ldots, v_k$; the **arcs** of this walk are $a_1, a_2, \ldots, a_k$. This walk is said to **start** at v_0 and **end** at v_k; it is also said to be a **walk from** v_0 **to** v_k. Its **starting point** is v_0, and its **ending point** is v_k. Its **length** is k.

(b) A **path** means a walk whose vertices are distinct.

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

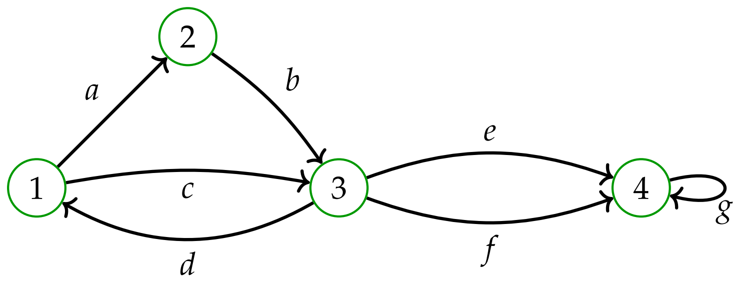
(d) A **cycle** means a closed walk $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ such that

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

(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}.)

Example 4.5.5. Consider the multidigraph

$D =$



.

Then, $(1, a, 2, b, 3, d, 1)$ and $(3, d, 1, c, 3)$ and $(4, g, 4)$ are three cycles of D, whereas $(3, d, 1, a, 2, b, 3, d, 1, c, 3)$ is a circuit but not a cycle.

4.5.2. Basic properties

Now, let us see which properties of walks, paths, closed walks and cycles remain valid for digraphs.

In Proposition 2.9.3, we saw how two walks in a simple graph could be combined ("spliced together") if the ending point of the first is the starting point of the second. In Proposition 3.3.7, we generalized this to multigraphs. The same holds for multidigraphs:

Proposition 4.5.6. Let D be a multidigraph. Let u, v and w be three vertices of D. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ be a walk from u to v. Let $\mathbf{b} = (b_0, f_1, b_1, \ldots, f_\ell, b_\ell)$ be a walk from v to w. Then,

$$\begin{aligned} & (a_0, e_1, a_1, \ldots, e_k, a_k, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \\ & = (a_0, e_1, a_1, \ldots, a_{k-1}, e_k, b_0, f_1, b_1, \ldots, f_\ell, b_\ell) \\ & = (a_0, e_1, a_1, \ldots, a_{k-1}, e_k, v, f_1, b_1, \ldots, f_\ell, b_\ell) \end{aligned}$$

is a walk from u to w. This walk shall be denoted $\mathbf{a} * \mathbf{b}$.

Proof. The same (trivial) argument as for undirected graphs works here. ☐

However, unlike for undirected graphs, we can no longer reverse walks or paths in digraphs. Thus, it often happens that there is a walk from u to v, but no walk from v to u.

Reducing a walk to a path (as we did in Proposition 2.9.5 for simple graphs and in Proposition 3.3.9 for multigraphs) still works for multidigraphs:

Proposition 4.5.7. Let D be a multidigraph. Let u and v be two vertices of D. Let $\mathbf{a}$ be a walk from u to v. Let k be the length of $\mathbf{a}$. Assume that $\mathbf{a}$ is not a path. Then, there exists a walk from u to v whose length is smaller than k.

Corollary 4.5.8 (When there is a walk, there is a path)**.** Let D be a multidigraph. Let u and v be two vertices of D. 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 $\leq k$.

The proofs of these facts are the same as for multigraphs.

The following proposition is an analogue of Proposition 2.10.4 for multidigraphs:

Proposition 4.5.9. Let D be a multidigraph. Let $\mathbf{w}$ be a walk of D. Then, $\mathbf{w}$ either is a path or contains a cycle (i.e., there exists a cycle of D whose arcs are arcs of $\mathbf{w}$).

Proof. This follows by the same argument as Proposition 2.10.4. □

Given a multidigraph D and two vertices u and v of D, we can pose the same five algorithmic questions (Questions 1, 2, 3, 4 and 5) that we posed for a simple graph G in Subsection 2.9.4. As with multigraphs, the same answers that we gave back then are still valid in our new setting, as long as we replace "neighbors of v" by "in-neighbors of v" (that is, vertices w such that D has an arc from w to v), and as long as we keep track of the arcs in our paths or walks.

4.5.3. Exercises

Exercise 4.1. Let D be a multidigraph with at least one vertex. Prove the following:

(a) If each vertex v of D satisfies $\deg^+ v > 0$, then D has a cycle.

(b) If each vertex v of D satisfies $\deg^+ v = \deg^- v = 1$, then each vertex of D belongs to exactly one cycle of D. Here, two cycles are considered to be identical if one can be obtained from the other by cyclic rotation.

Exercise 4.2. Let p be a prime number. Let $(a_1, a_2, a_3, \ldots)$ be a sequence of integers that is periodic with period p (that is, that satisfies $a_i = a_{i+p}$ for each $i > 0$). Assume that $a_1 + a_2 + \cdots + a_p$ is not divisible by p. Prove that there exists an $i \in \{1, 2, \ldots, p\}$ such that none of the p numbers

$$a_i, \ a_i + a_{i+1}, \ a_i + a_{i+1} + a_{i+2}, \ \ldots, \ a_i + a_{i+1} + \cdots + a_{i+p-1}$$

(that is, of the p sums $a_i + a_{i+1} + \cdots + a_j$ for $i \leq j < i + p$) is divisible by p.

[**Remark:** This would be false if p was not prime. For instance, for $p = 4$, the sequence $(0, 2, 2, 2, 0, 2, 2, 2, \ldots)$ would be a counterexample.]

[**Hint:** Use Exercise 4.1 **(a)**. What is the digraph, and why does it have a cycle?]

Exercise 4.3. Let $D = (V, A, \psi)$ be a multidigraph.

For two vertices u and v of D, we shall write $u \overset{*}{\to} v$ if there exists a path from u to v.

A **root** of D means a vertex $u \in V$ such that each vertex $v \in V$ satisfies $u \overset{*}{\to} v$.

A **common ancestor** of two vertices u and v means a vertex $w \in V$ such that $w \overset{*}{\to} u$ and $w \overset{*}{\to} v$.

Assume that D has at least one vertex. Prove that D has a root if and only if every two vertices in D have a common ancestor.

The following exercise is both a directed analogue and a generalization of Mantel's theorem (Theorem 2.4.6):

Exercise 4.4. Let D be a simple digraph with n vertices and a arcs. Assume that D has no loops, and that we have $a > n^2/2$. Prove the following:

(a) The digraph D has a cycle of length 3.

(b) We define an **enhanced** 3**-cycle** to be a triple (u, v, w) of distinct vertices of D such that all four pairs (u, v), (v, w), (w, u) and (u, w) are arcs of D. Then, the digraph D has an enhanced 3-cycle.

Exercise 4.5. Let $D = (V, A)$ be a simple digraph that has no cycles.

If $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is a list of vertices of D (not necessarily a walk!), then a **back-cut** of $\mathbf{v}$ shall mean an arc $a \in A$ whose source is v_i and whose target is v_j for some $i, j \in \{1, 2, \ldots, n\}$ satisfying $i > j$. (Colloquially speaking, a back-cut of $\mathbf{v}$ is an arc of D that leads from some vertex of $\mathbf{v}$ to some earlier vertex of $\mathbf{v}$.)

A list $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ of vertices of D is said to be a **toposort**[24] of D if it contains each vertex of D exactly once and has no back-cuts.

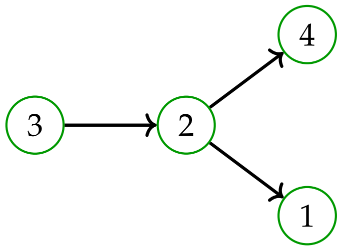
Prove the following:

(a) The digraph D has at least one toposort.

(b) If D has only one toposort, then this toposort is a Hamiltonian path of D.

Here, a **Hamiltonian path** in D means a walk of D that contains each vertex of D exactly once.

[**Example:** For example, the digraph



has two toposorts: $(3, 2, 1, 4)$ and $(3, 2, 4, 1)$.]

Exercise 4.6. Let n be a positive integer. Let D be a digraph that has no cycles of length ≤ 2. Assume that D has at least 2^{n-1} vertices. Prove that D has an induced subdigraph that has n vertices and has no cycles.

4.5.4. The adjacency matrix

A simple way to find the number of walks from a given vertex to a given vertex in a multidigraph is provided by matrix algebra:

Theorem 4.5.10. Let $D = (V, A, \psi)$ be a multidigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$.

If M is any matrix, and if i and j are two positive integers, then $M_{i,j}$ shall denote the (i, j)-th entry of M (that is, the entry of M in the i-th row and the j-th column).

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

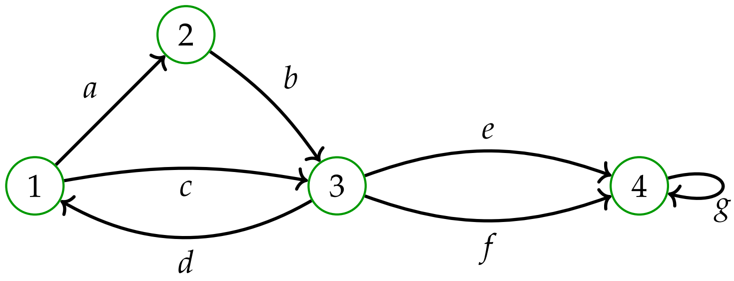
$$C_{i,j} = (\text{the number of all arcs } a \in A \text{ with source } i \text{ and target } j) \qquad \text{for all } i, j \in V.$$

Let $k \in \mathbb{N}$, and let $i, j \in V$. Then, $\left(C^k\right)_{i,j}$ equals the number of all walks of D having starting point i, ending point j and length k.

Remark 4.5.11. The matrix C in Theorem 4.5.10 is known as the **adjacency**

[24]This is short for "topological sorting". I don't know where this name comes from.

matrix of D. For example, if the multidigraph is

$D =$ 

then its adjacency matrix is

$$C = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and thus Theorem 4.5.10 yields (among other things) that the $(1,3)$-rd entry $\left(C^k\right)_{1,3}$ of its k-th power C^k equals the number of all walks of D having starting point 1, ending point 3 and length k.

The adjacency matrix of a multidigraph D determines D up to the identities of the arcs, and thus is often used as a convenient way to encode a multidigraph.

Proof of Theorem 4.5.10. Forget that we fixed i, j and k. We want to prove the following claim:

Claim 1: Let $i \in V$ and $j \in V$ and $k \in \mathbb{N}$. Then,

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

Before we prove this claim, let us recall that C is the adjacency matrix of D. Thus, for each $i \in V$ and $j \in V$, we have

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

(by the definition of the adjacency matrix). In other words, for each $i \in V$ and $j \in V$, we have

$$C_{i,j} = (\text{the number of arcs from } i \text{ to } j),$$

where we agree that an "arc from i to j" means an arc $a \in A$ with source i and target j.

Renaming i as w in this statement, we obtain the following: For each $w \in V$ and $j \in V$, we have

$$C_{w,j} = (\text{the number of arcs from } w \text{ to } j). \tag{6}$$

Let us also recall that any two $n \times n$-matrices M and N satisfy

$$(MN)_{i,j} = \sum_{w=1}^{n} M_{i,w} N_{w,j} \tag{7}$$

for any $i \in V$ and $j \in V$. (Indeed, this is just the rule for how matrices are multiplied.)

We can now prove Claim 1:

[*Proof of Claim 1:* We shall prove Claim 1 by induction on k:

Induction base: We shall first prove Claim 1 for $k = 0$.

Indeed, let $i \in V$ and $j \in V$. The 0-th power of any $n \times n$-matrix is defined to be the $n \times n$ identity matrix I_n; thus, $C^0 = I_n$. Hence,

$$\left(C^0\right)_{i,j} = (I_n)_{i,j} = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{if } i \neq j \end{cases} \tag{8}$$

(by the definition of the identity matrix).

On the other hand, how many walks from i to j have length 0 ? A walk that has length 0 must consist of a single vertex, which is simultaneously the starting point and the ending point of this walk. Thus, a walk from i to j that has length 0 exists only when $i = j$, and in this case there is exactly one such walk (namely, the walk (i)). Hence,

$$(\text{the number of walks from } i \text{ to } j \text{ that have length } 0) = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{if } i \neq j. \end{cases}$$

Comparing this with (8), we conclude that

$$\left(C^0\right)_{i,j} = (\text{the number of walks from } i \text{ to } j \text{ that have length } 0). \tag{9}$$

Now, forget that we fixed i and j. We thus have proven (9) for any $i \in V$ and $j \in V$. In other words, Claim 1 holds for $k = 0$. Thus, the induction base is complete.

Induction step: Let g be a positive integer. Assume that Claim 1 holds for $k = g - 1$. We must show that Claim 1 holds for $k = g$ as well.

We have assumed that Claim 1 holds for $k = g - 1$. In other words, for any $i \in V$ and $j \in V$, we have

$$\left(C^{g-1}\right)_{i,j} = (\text{the number of walks from } i \text{ to } j \text{ that have length } g - 1).$$

Renaming j as w in this statement, we obtain the following: For any $i \in V$ and $w \in V$, we have

$$\left(C^{g-1}\right)_{i,w} = (\text{the number of walks from } i \text{ to } w \text{ that have length } g - 1). \tag{10}$$

Each walk from i to j that has length g has the form

$$\mathbf{w} = \left(v_0, a_1, v_1, a_2, v_2, \ldots, a_{g-1}, v_{g-1}, a_g, v_g\right)$$

for some vertices $v_0, v_1, \ldots, v_g$ of D and some arcs $a_1, a_2, \ldots, a_g$ of D satisfying $v_0 = i$ and $v_g = j$ and $\left(\psi\left(a_h\right) = \left(v_{h-1}, v_h\right) \text{ for all } h \in \{1, 2, \ldots, g\}\right)$. Thus, each such walk $\mathbf{w}$ can be constructed by the following algorithm:

- First, we choose a vertex w of D to serve as the vertex v_{g-1} (that is, as the penultimate vertex of the walk $\mathbf{w}$). This vertex w must belong to V.
- Now, we choose the vertices $v_0, v_1, \ldots, v_{g-1}$ (that is, all vertices of our walk except for the last one) and the arcs $a_1, a_2, \ldots, a_{g-1}$ (that is, all arcs of our walk except for the last one) in such a way that $v_{g-1} = w$. This is tantamount to choosing a walk $\left(v_0, a_1, v_1, a_2, v_2, \ldots, a_{g-1}, v_{g-1}\right)$ from i to w that has length $g-1$. This choice can be made in $\left(C^{g-1}\right)_{i,w}$ many ways (because (10) shows that the number of walks from i to w that have length $g-1$ is $\left(C^{g-1}\right)_{i,w}$).
- We have now determined all but the last vertex and all but the last arc of our walk $\mathbf{w}$. We set the last vertex v_g of our walk to be j. (This is the only possible option, since our walk $\mathbf{w}$ has to be a walk from i to j.)
- We choose the last arc a_g of our walk $\mathbf{w}$. This arc a_g must have source v_{g-1} and target v_g; in other words, it must have source w and target j (since $v_{g-1} = w$ and $v_g = j$). In other words, it must be an arc from w to j. Thus, it can be chosen in $C_{w,j}$ many ways (because (6) shows that the number of arcs from w to j is $C_{w,j}$).

Conversely, of course, this algorithm always constructs a walk from i to j that has length g, and different choices in the algorithm lead to distinct walks. Thus, the total number of walks from i to j that have length g equals the total number of choices in the algorithm. But the latter number is $\sum\limits_{w \in V} \left(C^{g-1}\right)_{i,w} C_{w,j}$ (since the algorithm first chooses a $w \in V$, then involves a step with $\left(C^{g-1}\right)_{i,w}$ choices, and then involves a step with $C_{w,j}$ choices). Hence, the total number of walks from i to j that have length g is $\sum\limits_{w \in V} \left(C^{g-1}\right)_{i,w} C_{w,j}$. In other words,

$$(\text{the number of walks from } i \text{ to } j \text{ that have length } g) = \sum_{w \in V} \left(C^{g-1}\right)_{i,w} C_{w,j}.$$

Comparing this with

$$\Big(\underbrace{C^g}_{=C^{g-1}C}\Big)_{i,j} = \left(C^{g-1}C\right)_{i,j} = \sum_{w=1}^{n} \left(C^{g-1}\right)_{i,w} C_{w,j}$$
$$\left(\text{by (7) (applied to } M = C^{g-1} \text{ and } N = C)\right)$$
$$= \sum_{w \in V} \left(C^{g-1}\right)_{i,w} C_{w,j} \qquad \left(\text{since } \{1,2,\ldots,n\} = V\right),$$

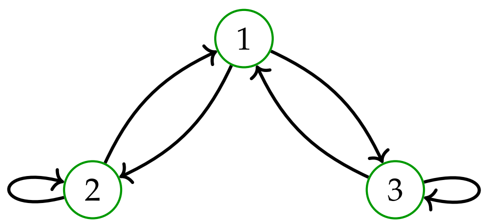
we obtain

$$\left(C^g\right)_{i,j} = (\text{the number of walks from } i \text{ to } j \text{ that have length } g). \tag{11}$$

Now, forget that we fixed i and j. We thus have proven (11) for any $i \in V$ and $j \in V$. In other words, Claim 1 holds for $k = g$. Thus, the induction step is complete. Hence, Claim 1 is proven by induction.]

Theorem 4.5.10 follows immediately from Claim 1. $\square$

Exercise 4.7. Let E be the following multidigraph:

$E =$ 

Let $n \in \mathbb{N}$. Compute the number of walks from 1 to 1 having length n.

4.6. Connectedness strong and weak

We defined the "path-connected" relation for undirected graphs using the existence of paths (see Definition 2.9.8). For a digraph, however, the relations "there is a walk from u to v" and "there is a walk from v to u" are (in general) distinct and non-symmetric, so I prefer not to give them a symmetric-looking symbol such as $\simeq_D$. Instead, we define **strong path-connectedness** to mean the existence of **both** walks:

Definition 4.6.1. 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 in D and there exists a walk from v to u in D.

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

Example 4.6.2. Let D be as in Example 4.5.5. Then, $1 \simeq_D 2$, because there exists a walk from 1 to 2 in D (for instance, $(1, a, 2)$) and there also exists a walk from 2 to 1 in D (for instance, $(2, b, 3, d, 1)$). However, we don't have $3 \simeq_D 4$. Indeed, while there exists a walk from 3 to 4 in D, there exists no walk from 4 to 3 in D.

Proposition 4.6.3. Let D be a multidigraph. Then, the relation $\simeq_D$ is an equivalence relation.

Proof. Easy, like for simple graphs. □

Again, we can replace "walk" by "path" in the definition of the relation $\simeq_D$:

Proposition 4.6.4. Let D be a multidigraph. Let u and v be two vertices of D. Then, $u \simeq_D v$ if and only if there exist a path from u to v and a path from v to u.

Proof. Easy, like for simple graphs. □

Definition 4.6.5. Let D be a multidigraph. The equivalence classes of the equivalence relation $\simeq_D$ are called the **strong components** of D.

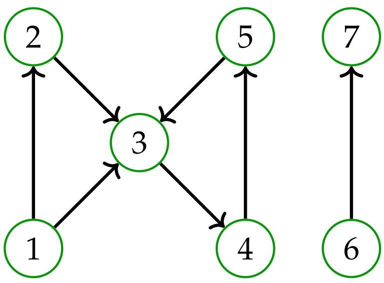
Definition 4.6.6. Let D be a multidigraph. We say that D is **strongly connected** if D has exactly one strong component.

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.[25]

In comparison, here is a weaker notion of connected components and connectedness:

Definition 4.6.7. Let D be a multidigraph. Consider its underlying undirected multigraph D^{und}. The components of this undirected multigraph D^{und} (that is, the equivalence classes of the equivalence relation $\simeq_{D^{\text{und}}}$) are called the **weak components** of D. We say that D is **weakly connected** if D has exactly one weak component (i.e., if D^{und} is connected).

Example 4.6.8. Let D be the following simple digraph:

$D =$ 

.

[25] Some authors use the word "**diconnected**" for "strongly connected". As this word is just a single letter away from "disconnected", I cannot recommend it.

We treat D as a multidigraph (namely, D^{mult}).

The weak components of D are $\{1,2,3,4,5\}$ and $\{6,7\}$.

The strong components of D are $\{1\}$, $\{2\}$, $\{3,4,5\}$, $\{6\}$ and $\{7\}$. (Indeed, for example, we have $1 \not\simeq_D 2 \not\simeq_D 3$ but $3 \simeq_D 4 \simeq_D 5$.)

So D is neither strongly nor weakly connected, but has more strong than weak components.

Example 4.6.9. The digraph from Example 4.5.2 is weakly connected, but not at all strongly connected (indeed, each of its strong components has size 1). The digraph from Example 4.5.3, on the other hand, is strongly connected.

Proposition 4.6.10. Any strongly connected digraph is weakly connected.

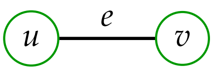
Proof. Let D be a multidigraph. Then, any walk of D is (or, more precisely, gives rise to) a walk of D^{und}. Hence, if two vertices u and v of D are strongly path-connected in D, then they are path-connected in D^{und}. Therefore, if D is strongly connected, then D^{und} is connected, but this means that D is weakly connected. ☐

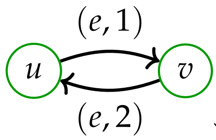
Exercise 4.8. Let D be a multidigraph. Prove that the strong components of D are the weak components of D if and only if each arc of D is contained in at least one cycle.

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

Proposition 4.6.11. Let G be a multigraph. Then:

- **(a)** The walks of G are "more or less the same as" the walks of the multidigraph G^{bidir}. More precisely, each walk of G gives rise to a walk of G^{bidir} (with the same starting point and the same ending point), and conversely, each walk of G^{bidir} gives rise to a walk of G. If G has no loops, then this is a one-to-one correspondence (i.e., a bijection) between the walks of G and the walks of G^{bidir}.
- **(b)** The paths of G are "more or less the same as" the paths of the multidigraph G^{bidir}. This is always a one-to-one correspondence, since paths cannot contain loops.
- **(c)** The closed walks of G are "more or less the same as" the closed walks of the multidigraph G^{bidir}.
- **(d)** The cycles of G are not quite the same as the cycles of G^{bidir}. In fact, if e is an edge of G with two distinct 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} (this is best seen on a picture: G has the edge 

whereas G^{bidir} has the arc-pair 

), so G^{bidir} usually has more cycles than G has. But it is true that each cycle of G gives rise to a cycle of G^{bidir}.

Exercise 4.9. Let $D = (V, E, \psi)$ be a multidigraph.

Let A, B and C be three subsets of V such that the induced subdigraphs $D[A]$, $D[B]$ and $D[C]$ are strongly connected.

A cycle of D will be called **eclectic** if it contains at least one arc of $D[A]$, at least one arc of $D[B]$ and at least one arc of $D[C]$ (although these three arcs are not required to be distinct).

Prove the following:

(a) If the sets $B \cap C$, $C \cap A$ and $A \cap B$ are nonempty, but $A \cap B \cap C$ is empty, then D has an eclectic cycle.

(b) If the induced subdigraphs $D[B \cap C]$, $D[C \cap A]$ and $D[A \cap B]$ are strongly connected, but the induced subdigraph $D[A \cap B \cap C]$ is not strongly connected, then D has an eclectic cycle.

[**Note:** Keep in mind that the multidigraph with 0 vertices does not count as strongly connected.]

[**Solution:** This is a generalization of Exercise 7 on midterm #2 from my Spring 2017 course; see the course page for solutions.]

4.7. Eulerian walks and circuits

We have studied Eulerian walks and circuits for (undirected) multigraphs in Section 3.4. Let us now define analogous concepts for multidigraphs:

Definition 4.7.1. 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.

(In other words: A walk $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ of D is said to be **Eulerian** if for each arc a of D, there exists exactly one $i \in \{1, 2, \ldots, k\}$ such that $a = a_i$.)

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

The Euler–Hierholzer theorem gives a necessary and sufficient criterion for a multigraph to have an Eulerian circuit or walk. For multidigraphs, there is an analogous result:

Theorem 4.7.2 (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 of D satisfies $\deg^+ v = \deg^- v$.

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

Exercise 4.10. Prove Theorem 4.7.2.

Incidentally, the "each vertex v of D satisfies $\deg^+ v = \deg^- v$" condition has a name:

Definition 4.7.3. A multidigraph D is said to be **balanced** if each vertex v of D satisfies $\deg^+ v = \deg^- v$.

So balancedness is necessary and sufficient for the existence of an Eulerian circuit in a weakly connected multidigraph.

The following proposition is obvious:

Proposition 4.7.4. Let G be a multigraph. Then, the multidigraph G^{bidir} is balanced.

Proof. The definition of G^{bidir} yields that each vertex v of G^{bidir} satisfies $\deg^+ v = \deg v$ and $\deg^- v = \deg v$, where $\deg v$ denotes the degree of v as a vertex of G. Hence, each vertex v of G^{bidir} satisfies $\deg^+ v = \deg v = \deg^- v$. In other words, G^{bidir} is balanced. $\square$

Combining this proposition with Theorem 4.7.2 **(a)**, we can obtain a curious fact about undirected(!) multigraphs:

Theorem 4.7.5. 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.

Proof. The multidigraph G^{bidir} is balanced (by Proposition 4.7.4) and weakly connected (this follows easily from the connectedness of G). Hence, Theorem 4.7.2 **(a)** can be applied to $D = G^{\text{bidir}}$. Thus, G^{bidir} has an Eulerian circuit. Reinterpreting this circuit as a circuit of G, we obtain a circuit of G that contains each edge **exactly twice**, and uses it once in each direction. This proves Theorem 4.7.5. □

4.8. Hamiltonian cycles and paths

We can define Hamiltonian paths and cycles for simple digraphs in the same way as we defined them for simple graphs:

Definition 4.8.1. Let $D = (V, A)$ be a simple digraph.

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

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

Convention 4.8.2. In the following, we will abbreviate:

- "Hamiltonian path" as "**hamp**";
- "Hamiltonian cycle" as "**hamc**".

We might wonder what can be said about hamps and hamcs for digraphs. Is there an analogue of Ore's theorem? The answer is "yes", but it is significantly harder to prove:

Theorem 4.8.3 (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 vertices u and v satisfying $u \neq v$ and $(u, v) \notin A$ and $(v, u) \notin A$, we have $\deg u + \deg v \geq 2n - 1$. Here, $\deg w$ means $\deg^+ w + \deg^- w$. Then, D has a hamc.

For the (rather complicated) proof of this, see [BonTho77] or [Berge91, §10.3, Theorem 7]. Note that the "strongly connected" condition is needed.

4.9. The reverse and complement digraphs

We take a break from studying hamps (Hamiltonian paths) in order to introduce two more operations on simple digraphs.

Definition 4.9.1. 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).

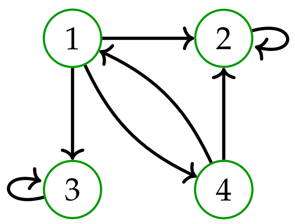
(c) We define D^{rev} as the simple digraph (V, A^{rev}), where

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

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

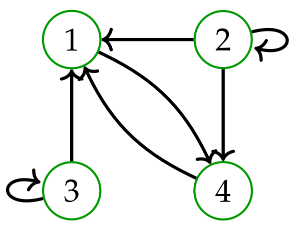
(d) We define $\overline{D}$ as the simple digraph $(V, (V \times V) \setminus A)$. This is the digraph that has the same vertices as D, but whose arcs are precisely the non-arcs of D. This digraph $\overline{D}$ is called the **complement** of D.

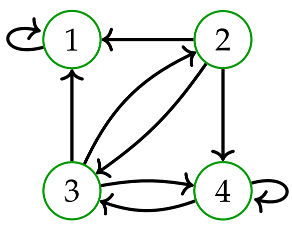
Example 4.9.2. Let

$D =$ 

.

Then,

$D^{\text{rev}} =$ 

and $\overline{D} =$ 

.

Convention 4.9.3. In the following, the symbol # means "number". For example,

$$(\# \text{ of subsets of } \{1, 2, 3\}) = 8.$$

We now shall try to count hamps in simple digraphs[26]. As a warmup, here is a particularly simple case:

[26]See [17s-lec7] for a more detailed treatment of this topic.

Proposition 4.9.4. Let D be the simple digraph (V, A), where

$$V = \{1, 2, \ldots, n\} \qquad \text{for some } n \in \mathbb{N},$$

and where

$$A = \{(i, j) \mid i < j\}.$$

Then, (# of hamps of D) $= 1$.

Proof. It is easy to see that the only hamp of D is $(1, 2, \ldots, n)$. □

The following is easy, too:

Proposition 4.9.5. Let D be a simple digraph. Then,

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

Proof. The hamps of D^{rev} are obtained from the hamps of D by walking backwards. □

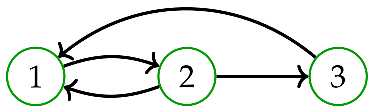
So far, so boring. What about this:

Theorem 4.9.6 (Berge's theorem)**.** Let D be a simple digraph. Then,

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

This is much less obvious or even expected. We first give an example:

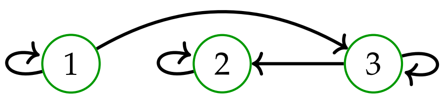
Example 4.9.7. Let D be the following digraph:

$D =$ 

.

This digraph has 3 hamps: $(1, 2, 3)$ and $(2, 3, 1)$ and $(3, 1, 2)$.

Its complement $\overline{D}$ looks as follows:

$\overline{D} =$ 

.

It has only 1 hamp: $(1, 3, 2)$.

Thus, in this case, Theorem 4.9.6 says that $1 \equiv 3 \bmod 2$.

Proof of Theorem 4.9.6. (This is an outline; see [17s-lec7, proof of Theorem 1.3.6] for more details.)

Write the simple digraph D as $D = (V, A)$, and assume WLOG that $V \neq \varnothing$. Set $n = |V|$.

A **V-listing** will mean a list of elements of V that contains each element of V exactly once. (Thus, each V-listing is an n-tuple, and there are $n!$ many V-listings.) Note that a V-listing is the same as a hamp of the "complete" digraph $(V, V \times V)$. Any hamp of D or of $\overline{D}$ is therefore a V-listing, but not every V-listing is a hamp of D or $\overline{D}$.

If $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a V-listing, then we define a set

$$P(\sigma) := \{\sigma_1\sigma_2,\ \sigma_2\sigma_3,\ \ldots,\ \sigma_{n-1}\sigma_n\}.$$

We call this set $P(\sigma)$ the **arc set** of σ. When we regard σ as a hamp of $(V, V \times V)$, this set $P(\sigma)$ is just the set of all arcs of σ. Note that this is an $(n-1)$-element set. We make a few easy observations (prove them!):

> *Observation 1:* We can reconstruct a V-listing σ from its arc set $P(\sigma)$. In other words, the map $\sigma \mapsto P(\sigma)$ is injective.
>
> *Observation 2:* Let σ be a V-listing. Then, σ is a hamp of D if and only if $P(\sigma) \subseteq A$.
>
> *Observation 3:* Let σ be a V-listing. Then, σ is a hamp of $\overline{D}$ if and only if $P(\sigma) \subseteq (V \times V) \setminus A$.

Now, let N be the # of pairs (σ, B), where σ is a V-listing and B is a subset of A satisfying $B \subseteq P(\sigma)$. Thus,

$$N = \sum_{\sigma \text{ is a } V\text{-listing}} N_\sigma,$$

where

$$N_\sigma = (\#\text{ of subsets } B \text{ of } A \text{ satisfying } B \subseteq P(\sigma)).$$

But we also have

$$N = \sum_{B \text{ is a subset of } A} N^B,$$

where

$$N^B = (\#\text{ of } V\text{-listings } \sigma \text{ satisfying } B \subseteq P(\sigma)).$$

Let us now relate these two sums to hamps. We begin with $\sum\limits_{\sigma \text{ is a } V\text{-listing}} N_\sigma$.

We shall use the **Iverson bracket notation**: i.e., the notation $[\mathcal{A}]$ for the truth value of a statement $\mathcal{A}$. This truth value is defined to be the number 1 if $\mathcal{A}$ is true, and 0 if $\mathcal{A}$ is false. For instance,

$$[2+2=4] = 1 \qquad \text{and} \qquad [2+2=5] = 0.$$

For any V-listing σ, we have

$$\begin{aligned}
N_\sigma &= (\text{\# of subsets } B \text{ of } A \text{ satisfying } B \subseteq P(\sigma)) \\
&= (\text{\# of subsets } B \text{ of } A \cap P(\sigma)) \\
&= 2^{|A \cap P(\sigma)|} \\
&\equiv [|A \cap P(\sigma)| = 0] \qquad (\text{since } 2^m \equiv [m = 0] \bmod 2 \text{ for each } m \in \mathbb{N}) \\
&= [A \cap P(\sigma) = \varnothing] \qquad \left(\begin{array}{c}\text{since equivalent statements have the} \\ \text{same truth value}\end{array}\right) \\
&= [P(\sigma) \subseteq (V \times V) \setminus A] \qquad (\text{since } P(\sigma) \text{ is always a subset of } V \times V) \\
&= \left[\sigma \text{ is a hamp of } \overline{D}\right] \bmod 2 \qquad (\text{by Observation 3}).
\end{aligned}$$

So

$$\begin{aligned}
N &= \sum_{\sigma \text{ is a } V\text{-listing}} \underbrace{N_\sigma}_{\equiv \left[\sigma \text{ is a hamp of } \overline{D}\right] \bmod 2} \\
&\equiv \sum_{\sigma \text{ is a } V\text{-listing}} \left[\sigma \text{ is a hamp of } \overline{D}\right] \\
&= \left(\text{\# of } V\text{-listings } \sigma \text{ that are hamps of } \overline{D}\right) \\
&\qquad \left(\begin{array}{c}\text{because } \sum_{\sigma \text{ is a } V\text{-listing}} \left[\sigma \text{ is a hamp of } \overline{D}\right] \text{ is a sum} \\ \text{of several 1's and several 0's, and the 1's in this} \\ \text{sum correspond precisely to} \\ \text{the } V\text{-listings } \sigma \text{ that are hamps of } \overline{D}\end{array}\right) \\
&= \left(\text{\# of hamps of } \overline{D}\right) \bmod 2.
\end{aligned}$$

What about the other expression for N ? Recall that

$$N = \sum_{B \text{ is a subset of } A} N^B,$$

where

$$N^B = (\text{\# of } V\text{-listings } \sigma \text{ satisfying } B \subseteq P(\sigma)).$$

We want to prove that this sum equals (# of hamps of D), at least modulo 2.

So let B be a subset of A. We want to know $N^B \bmod 2$. In other words, we want to know when N^B is odd.

Let us first assume that N^B is odd, and see what follows from this.

Since N^B is odd, we have $N^B > 0$. Thus, there exists **at least one** V-listing σ satisfying $B \subseteq P(\sigma)$. We shall now draw some conclusions from this.

First, a definition: A **path cover** of V means a set of paths in the "complete" digraph $(V, V \times V)$ such that each vertex $v \in V$ is contained in exactly one of these paths. The **set of arcs** of such a path cover is simply the set of all arcs of all its paths. For example, if $V = \{1, 2, 3, 4, 5, 6, 7\}$, then

$$\{(1,3,5),\ (2),\ (6),\ (7,4)\}$$

is a path cover of V, and its set of arcs is $\{13,\ 35,\ 74\}$.

Now, ponder the following: If we remove an arc $v_i v_{i+1}$ from a path $(v_1, v_2, \ldots, v_k)$, then this path breaks up into two paths $(v_1, v_2, \ldots, v_i)$ and $(v_{i+1}, v_{i+2}, \ldots, v_k)$. Thus, if we remove some arcs from the arc set $P(\sigma)$ of a V-listing σ, then we obtain the set of arcs of a path cover of V. (For instance, removing the arcs 52, 26 and 67 from the arc set $P(\sigma)$ of the V-listing $\sigma = (1,3,5,2,6,7,4)$ yields precisely the path cover $\{(1,3,5),\ \ (2),\ \ (6),\ \ (7,4)\}$ that we just showed as an example.)

Now, recall that there exists **at least one** V-listing σ satisfying $B \subseteq P(\sigma)$. Hence, B is obtained by removing some arcs from the arc set $P(\sigma)$ of this V-listing σ. Therefore, B is the set of arcs of a path cover of V (by the claim of the preceding paragraph). Let us say that this path cover consists of exactly r paths. Then,

$$(\text{\# of } V\text{-listings } \sigma \text{ satisfying } B \subseteq P(\sigma)) = r!,$$

because any such V-listing σ can be constructed by concatenating the r paths in our path cover in some order (and there are $r!$ possible orders).

Thus, $N^B = (\text{\# of } V\text{-listings } \sigma \text{ satisfying } B \subseteq P(\sigma)) = r!$. But we have assumed that N^B is odd. So $r!$ is odd. Since r is positive (because $V \neq \varnothing$, so our path cover must contain at least one path), this entails that $r = 1$. So our path cover is just a single path; this path is a path of D (since its set of arcs B is a subset of A) and therefore is a hamp of D (since it constitutes a path cover of V all by itself). If we denote it by σ, then we have $B = P(\sigma)$ (since B is the set of arcs of the path cover that consists of σ alone).

Forget our assumption that N^B is odd. We have thus shown that if N^B is odd, then $B = P(\sigma)$ for some hamp σ of D.

Conversely, it is easy to see that if $B = P(\sigma)$ for some hamp σ of D, then N^B is odd (and actually equals 1).

Combining these two results, we see that N^B is odd **if and only if** $B = P(\sigma)$ for some hamp σ of D. Therefore,

$$\left[N^B \text{ is odd}\right] = [B = P(\sigma) \ \text{ for some hamp } \sigma \text{ of } D].$$

However,

$$\begin{aligned} N^B &\equiv \left[N^B \text{ is odd}\right] && (\text{since } m \equiv [m \text{ is odd}] \bmod 2 \text{ for any } m \in \mathbb{Z}) \\ &= [B = P(\sigma) \ \text{ for some hamp } \sigma \text{ of } D] \bmod 2. \end{aligned}$$

We have proved this congruence for every subset B of A. Thus,

$$\begin{aligned}
N &= \sum_{B \text{ is a subset of } A} \underbrace{N^B}_{\equiv [B = P(\sigma) \text{ for some hamp } \sigma \text{ of } D] \bmod 2} \\
&\equiv \sum_{B \text{ is a subset of } A} [B = P(\sigma) \text{ for some hamp } \sigma \text{ of } D] \\
&= (\text{\# of subsets } B \text{ of } A \text{ such that } B = P(\sigma) \text{ for some hamp } \sigma \text{ of } D) \\
&= (\text{\# of sets of the form } P(\sigma) \text{ for some hamp } \sigma \text{ of } D) \\
&\qquad \left(\begin{array}{c}\text{because each set of the form } P(\sigma) \text{ for some} \\ \text{hamp } \sigma \text{ of } D \text{ is a subset of } A \text{ (by Observation 2)}\end{array}\right) \\
&= (\text{\# of hamps of } D) \bmod 2
\end{aligned}$$

(indeed, Observation 1 shows that different hamps σ have different sets $P(\sigma)$, so counting the sets $P(\sigma)$ for all hamps σ is equivalent to counting the hamps σ themselves).

Now we have proved that $N \equiv (\text{\# of hamps of } \overline{D}) \bmod 2$ and $N \equiv (\text{\# of hamps of } D) \bmod 2$. Comparing these two congruences, we obtain

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

This proves Berge's theorem. □

4.10. Tournaments

4.10.1. Definition

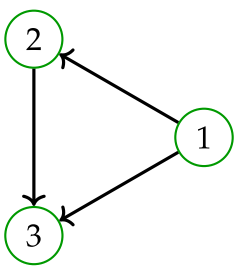
We now introduce a special class of simple digraphs.

Definition 4.10.1. A digraph D is said to be **loopless** if it has no loops.

Definition 4.10.2. A **tournament** is defined to be a loopless simple digraph D that satisfies the

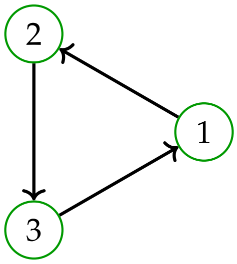
- **Tournament axiom:** For any two distinct vertices u and v of D, **exactly** one of (u, v) and (v, u) is an arc of D.

Example 4.10.3. The following digraph is a tournament:



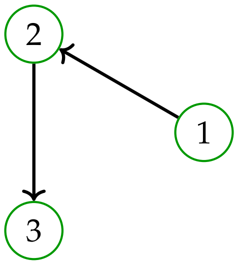
.

The following digraph is a tournament as well:



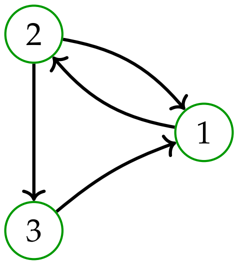
.

However, the following digraph is not a tournament:



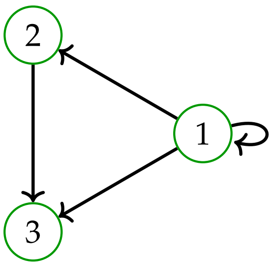
,

because the tournament axiom is not satisfied for $u = 1$ and $v = 3$. Nor is the following digraph a tournament:



,

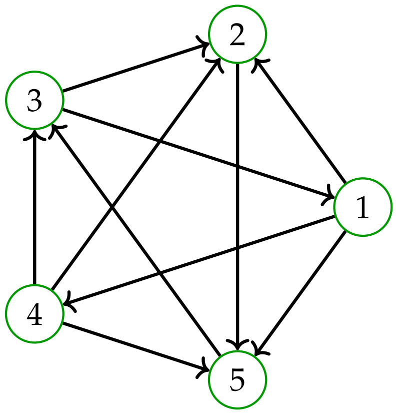
because the tournament axiom is not satisfied for $u = 1$ and $v = 2$. Finally, the digraph



is not a tournament either, since it is not loopless.

The digraph D in Proposition 4.9.4 always is a tournament.

Example 4.10.4. Here is a tournament with 5 vertices:



.

A tournament can also be viewed as a complete graph, whose each edge has been given a direction.

Using Definition 4.9.1, we can restate the definition of a tournament as follows:

Proposition 4.10.5. Let $D = (V, A)$ be a loopless simple digraph. Then, D is a tournament if and only if the non-loop arcs of $\overline{D}$ are precisely the arcs of D^{rev}.

Proof. Easy consequence of definitions. $\square$

Exercise 4.11. Let D be a tournament with at least one vertex.

We say that a vertex u of D **directly owns** a vertex w of D if (u, w) is an arc of D.

We say that a vertex u of D **indirectly owns** a vertex w of D if there exists a vertex v of D such that both (u, v) and (v, w) are arcs of D.

Prove that D has a vertex that (directly or indirectly) owns all other vertices.

[**Solution:** This exercise appears in [20f, Exercise 6.3.1] (restated in the language of players and matches) and in [Maurer80, Theorem 1] (restated in the language of chickens and pecking orders). It originates in a study of pecking orders by Landau [Landau53].]

4.10.2. The Rédei theorems

Which tournaments have hamps? The answer is surprisingly simple:[27]

[27] Here we agree to consider the empty list $()$ to be a hamp of the digraph $(\varnothing, \varnothing)$.

Theorem 4.10.6 (Easy Rédei theorem)**.** A tournament always has at least one hamp.

Even better, and perhaps even more surprisingly:

Theorem 4.10.7 (Hard Rédei theorem)**.** Let D be a tournament. Then,

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

Our goal now is to prove these two theorems. Clearly, the Easy Rédei Theorem follows from the Hard one, since an odd number cannot be 0. Thus, it will suffice to prove the Hard one.

The proof of the hard Rédei theorem will rely on the following crucial lemma:

Lemma 4.10.8. Let $D = (V, A)$ be a tournament, and let $vw \in A$ be an arc of D.

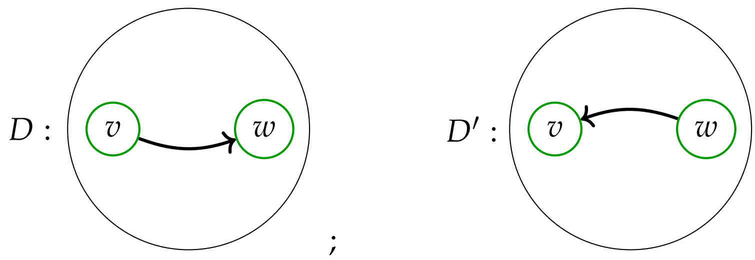
Let D' be the digraph obtained from D by reversing the arc vw. In other words, let

$$D' := (V, \ (A \setminus \{vw\}) \cup \{wv\}) .$$

Then, D' is again a tournament, and satisfies

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

Here is a visualization of the setup of Lemma 4.10.8:



; .

(Here, we are only showing the arcs joining v with w, since D and D' agree in all other arcs.)

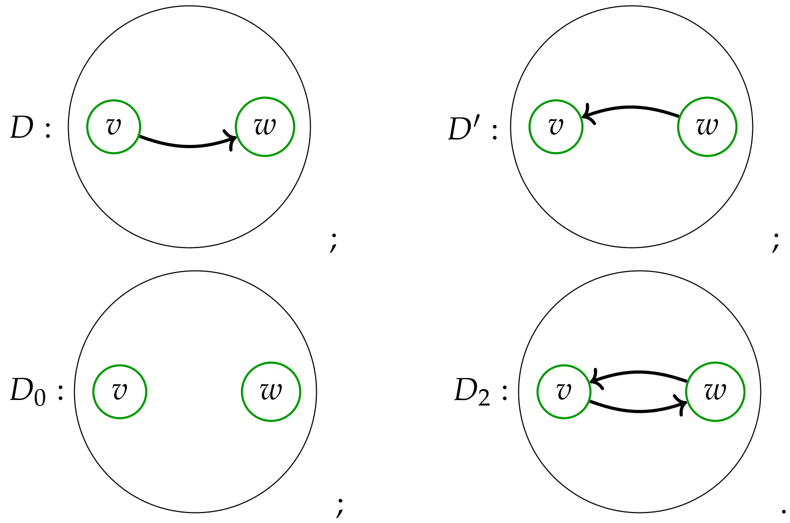
Proof of Lemma 4.10.8. (This is an outline; see [17s-lec7, proof of Lemma 1.6.2] for more details.)

First of all, D' is clearly a tournament. It remains to prove the congruence.

We introduce two more digraphs: Let

$$D_0 := (\text{the digraph } D \text{ with the arc } vw \text{ removed}) \qquad \text{and}$$
$$D_2 := (\text{the digraph } D \text{ with the arc } wv \text{ added}) .$$

Note that these are not tournaments any more. Here is a comparative illustration of all four digraphs D, D', D_0 and D_2 (again showing only the arcs joining v with w, since there are no differences in the other arcs):



The digraph D_0 is D' with the arc wv removed. Therefore, a hamp of D_0 is the same as a hamp of D' that does not use the arc wv. Hence,

$$\begin{aligned}
&(\#\text{ of hamps of } D_0) \\
&= (\#\text{ of hamps of } D' \text{ that do not use the arc } wv) \\
&= (\#\text{ of hamps of } D') - (\#\text{ of hamps of } D' \text{ that use the arc } wv) .
\end{aligned}$$

Similarly, since D is D_2 with the arc wv removed, we have

$$\begin{aligned}
&(\#\text{ of hamps of } D) \\
&= (\#\text{ of hamps of } D_2) - (\#\text{ of hamps of } D_2 \text{ that use the arc } wv) \\
&= (\#\text{ of hamps of } D_2) - (\#\text{ of hamps of } D' \text{ that use the arc } wv)
\end{aligned}$$

(the last equality is because a hamp of D_2 that uses the arc wv cannot use the arc vw, and therefore is automatically a hamp of D' as well, and of course the converse is obviously true).

However, from the previously proved equality

$$\begin{aligned}
&(\#\text{ of hamps of } D_0) \\
&= (\#\text{ of hamps of } D') - (\#\text{ of hamps of } D' \text{ that use the arc } wv) ,
\end{aligned}$$

we obtain

$$\begin{aligned}
&(\#\text{ of hamps of } D') \\
&= (\#\text{ of hamps of } D_0) + (\#\text{ of hamps of } D' \text{ that use the arc } wv) \\
&\equiv (\#\text{ of hamps of } D_0) - (\#\text{ of hamps of } D' \text{ that use the arc } wv) \bmod 2
\end{aligned}$$

(since $x + y \equiv x - y \mod 2$ for any integers x and y). Thus, if we can show that

$$(\text{\# of hamps of } D_2) \equiv (\text{\# of hamps of } D_0) \mod 2,$$

then we will be able to conclude that

$$\begin{aligned}
&(\text{\# of hamps of } D) \\
&= \underbrace{(\text{\# of hamps of } D_2)}_{\equiv (\text{\# of hamps of } D_0) \mod 2} - \left(\text{\# of hamps of } D' \text{ that use the arc } wv\right) \\
&\equiv (\text{\# of hamps of } D_0) - \left(\text{\# of hamps of } D' \text{ that use the arc } wv\right) \\
&\equiv \left(\text{\# of hamps of } D'\right) \mod 2,
\end{aligned}$$

and the proof of the lemma will be complete.

So let us show this. Recall that D is a tournament. Thus, the non-loop arcs of $\overline{D}$ are precisely the arcs of D^{rev} (by Proposition 4.10.5). Hence, the non-loop arcs of $\overline{D_0}$ are precisely the arcs of D_2^{rev} (since $\overline{D_0}$ is just $\overline{D}$ with the extra arc vw added, and since D_2^{rev} is just D^{rev} with the extra arc vw added). Therefore, the digraphs $\overline{D_0}$ and D_2^{rev} are equal "up to loops" (i.e., they have the same vertices and the same non-loop arcs). Since loops don't matter for hamps, these two digraphs thus have the same of hamps. Hence,

$$\left(\text{\# of hamps in } \overline{D_0}\right) = \left(\text{\# of hamps in } D_2^{\text{rev}}\right) = (\text{\# of hamps in } D_2)$$

(by Proposition 4.9.5), and therefore

$$(\text{\# of hamps in } D_2) = \left(\text{\# of hamps in } \overline{D_0}\right) \equiv (\text{\# of hamps in } D_0) \mod 2$$

(by Theorem 4.9.6). As explained above, this completes the proof of Lemma 4.10.8. □

Now, the Hard Rédei theorem has become easy:

Proof of Theorem 4.10.7. (This is an outline; see [17s-lec7, proof of Theorem 1.6.1] for more details.)

We need to prove that the # of hamps of D is odd. Lemma 4.10.8 tells us that the parity of this # does not change when we reverse a single arc of D. Thus, of course, if we reverse several arcs of D, then this parity does not change either. However, we can WLOG assume that the vertices of D are $1, 2, \ldots, n$ for some $n \in \mathbb{N}$, and then, by reversing the appropriate arcs, we can ensure that the arcs of D are

$$\begin{aligned}
&12,\ 13,\ 14,\ \ldots,\ 1n, \\
&\quad 23,\ 24,\ \ldots,\ 2n, \\
&\qquad \ldots, \\
&\qquad\qquad (n-1)\,n
\end{aligned}$$

(i.e., each arc of D has the form ij with $i < j$). But at this point, the tournament D has only one hamp: namely, $(1, 2, \ldots, n)$. So $(\# \text{ of hamps of } D) = 1$ is odd at this point. Since the parity of the # of hamps of D has not changed as we reversed our arcs, we thus conclude that it has always been odd. This proves the Hard Rédei theorem (Theorem 4.10.7). $\square$

As we already mentioned, the Easy Rédei theorem follows from the Hard Rédei theorem. But it also has a short self-contained proof ([17s-lec7, Theorem 1.4.9]).

Remark 4.10.9. Theorem 4.10.7 shows that the # of hamps in a tournament is an odd positive integer. Can it be any odd positive integer, or are certain odd positive integers impossible?

Surprisingly, 7 and 21 are impossible. All other odd numbers between 1 and 80555 are possible. For higher numbers, the answer is not known so far. See MathOverflow question #232751 ([MO232751]) for more details.

4.10.3. Hamiltonian cycles in tournaments

By the Easy Rédei theorem, every tournament has a hamp. But of course, not every tournament has a hamc[28]. One obstruction is clear:

Proposition 4.10.10. If a digraph D has a hamc, then D is strongly connected.

In general, this is only a necessary criterion for a hamc, not a sufficient one. Not every strongly connected digraph has a hamc. However, it turns out that for tournaments, it is also sufficient, as long as the tournament has enough vertices:

Theorem 4.10.11 (Camion's theorem)**.** If a tournament D is strongly connected and has at least two vertices, then D has a hamc.

Proof sketch. A detailed proof can be found in [17s-lec7, Theorem 1.5.5]; here is just a very rough sketch.

Let $D = (V, A)$ be a strongly connected tournament with at least two vertices.[29] We must show that D has a hamc.

It is easy to see that D has a cycle. Let $\mathbf{c} = (v_1, v_2, \ldots, v_k, v_1)$ be a cycle of maximum length. We shall show that $\mathbf{c}$ is a hamc.

Let C be the set $\{v_1, v_2, \ldots, v_k\}$ of all vertices of this cycle $\mathbf{c}$.

A vertex $w \in V \setminus C$ will be called a **to-vertex** if there exists an arc from some v_i to w.

[28] Recall that "**hamc**" is our shorthand for "Hamiltonian cycle".

[29] By the way, a tournament with exactly two vertices cannot be strongly connected (as it has only 1 arc). Thus, by requiring D to have at least two vertices, we have actually guaranteed that D has at least three vertices.

A vertex $w \in V \setminus C$ will be called a **from-vertex** if there exists an arc from w to some v_i.

Since D is a tournament, each vertex in $V \setminus C$ is a to-vertex or a from-vertex. In theory, a vertex could be both (having an arc from some v_i and also an arc to some other v_j). However, this does not actually happen. To see why, argue as follows:

- If a to-vertex w has an arc from some v_i, then it must also have an arc from v_{i+1} [30] (because otherwise there would be an arc from w to v_{i+1}, and then we could make our cycle $\mathbf{c}$ longer by interjecting w between v_i and v_{i+1}; but this would contradict the fact that $\mathbf{c}$ is a cycle of maximum length).

- Iterating this argument, we see that if a to-vertex w has an arc from some v_i, then it must also have an arc from v_{i+1}, an arc from v_{i+2}, an arc from v_{i+3}, and so on; i.e., it must have an arc from each vertex of $\mathbf{c}$. Consequently, w cannot be a from-vertex. This shows that a to-vertex cannot be a from-vertex.

Let F be the set of all from-vertices, and let T be the set of all to-vertices. Then, as we have just shown, F and T are disjoint. Moreover, $F \cup T = V \setminus C$. Since a to-vertex cannot be a from-vertex, we furthermore conclude that any to-vertex has an arc from each vertex of $\mathbf{c}$ (otherwise, it would be a from-vertex), and that any from-vertex has an arc to each vertex of $\mathbf{c}$ (otherwise, it would be a to-vertex).

Next, we argue that there cannot be an arc from a to-vertex t to a from-vertex f. Indeed, if there was such an arc, then we could make the cycle $\mathbf{c}$ longer by interjecting t and f between (say) v_1 and v_2.

In total, we now know that every vertex of D belongs to one of the three disjoint sets C, F and T, and furthermore there is no arc from T to F, no arc from T to C, and no arc from C to F. Thus, there exists no walk from a vertex in T to a vertex in C (because there is no way out of T). This would contradict the fact that D is strongly connected, unless the set T is empty. Hence, T must be empty. Similarly, F must be empty. Since $F \cup T = V \setminus C$, this entails that $V \setminus C$ is empty, so that $V = C$. In other words, each vertex of D is on our cycle $\mathbf{c}$. Therefore, $\mathbf{c}$ is a hamc. This proves Camion's theorem. $\square$

4.10.4. Application of tournaments to the Vandermonde determinant

To wrap up the topic of tournaments, let me briefly discuss a curious application of their theory: a combinatorial proof of the Vandermonde determinant formula. See [17s-lec8] for the many details I'll be omitting.

Recall the Vandermonde determinant formula:

[30] Here, indices are periodic modulo k, so that v_{k+1} means v_1.

Theorem 4.10.12 (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

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

Then, its determinant is

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

There are many simple proofs of this theorem (e.g., a few on its ProofWiki page, which works with the transpose matrix). I will now outline a combinatorial one, using tournaments. This proof goes back to Ira Gessel's 1979 paper [Gessel79].

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

As a warmup, let's assume that we have some number $y_{(i,j)}$ given for each pair (i, j) of integers, and let's expand the product

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

The result is a sum of 8 products, one for each way to pluck an addend out of each of the three little sums:

$$\begin{aligned} & \left(y_{(1,2)} + y_{(2,1)} \right) \left(y_{(1,3)} + y_{(3,1)} \right) \left(y_{(2,3)} + y_{(3,2)} \right) \\ &= y_{(1,2)} y_{(1,3)} y_{(2,3)} + y_{(1,2)} y_{(1,3)} y_{(3,2)} + y_{(1,2)} y_{(3,1)} y_{(2,3)} + y_{(1,2)} y_{(3,1)} y_{(3,2)} \\ &\qquad + y_{(2,1)} y_{(1,3)} y_{(2,3)} + y_{(2,1)} y_{(1,3)} y_{(3,2)} + y_{(2,1)} y_{(3,1)} y_{(2,3)} + y_{(2,1)} y_{(3,1)} y_{(3,2)}. \end{aligned}$$

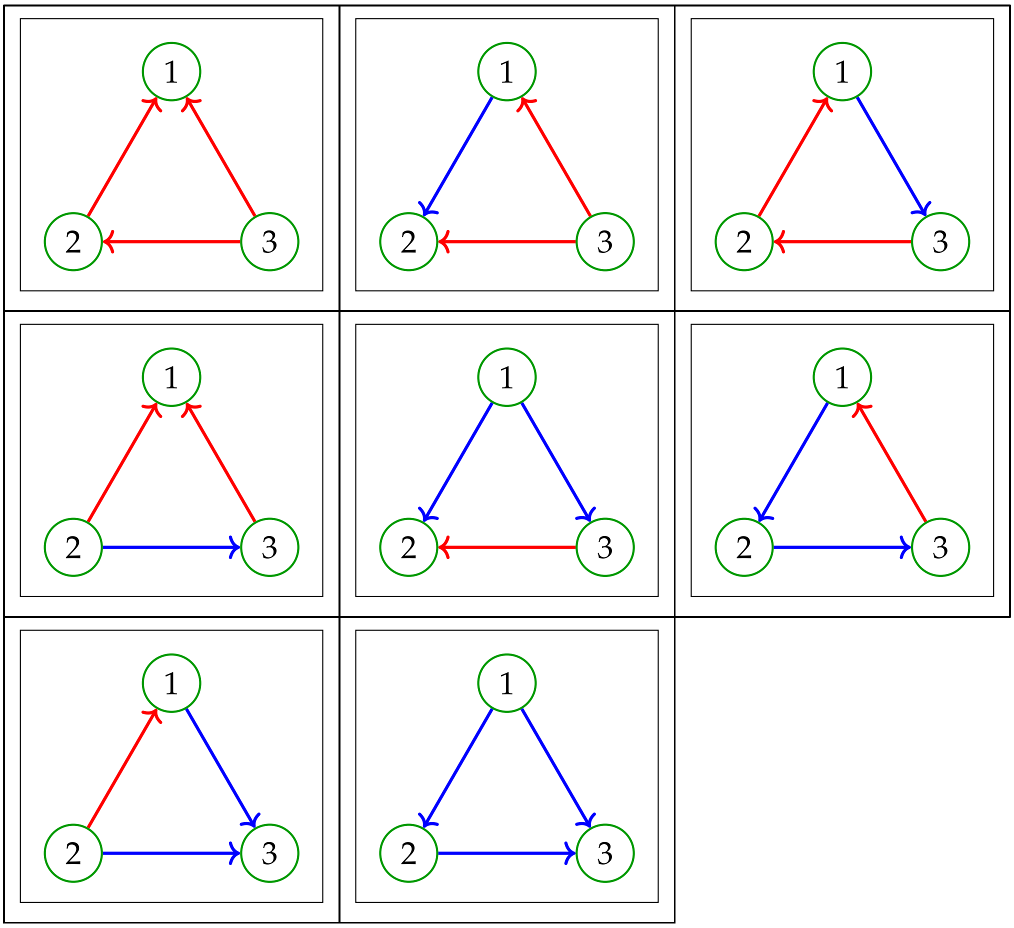
Note that each of the 8 products obtained has the form $y_a y_b y_c$, where

- a is one of the pairs $(1, 2)$ and $(2, 1)$,
- b is one of the pairs $(1, 3)$ and $(3, 1)$, and
- c is one of the pairs $(2, 3)$ and $(3, 2)$.

We can view these pairs a, b and c as the arcs of a tournament with vertex set $\{1, 2, 3\}$. Thus, our above expansion can be rewritten more compactly as follows:

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

For reference, here are all the 8 tournaments with vertex set $\{1, 2, 3\}$:



.

Here, for convenience, we are drawing an arc ij in blue if $i < j$ and in red otherwise.

This expansion can be generalized: We have

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

Substituting $y_{(i,j)} = \begin{cases} x_j, & \text{if } i < j; \\ -x_j, & \text{if } i \geq j \end{cases}$ in this equality, we obtain

$$\prod_{1 \leq i < j \leq n} (x_j - x_i) = \sum_{\substack{D \text{ is a tournament} \\ \text{with vertex set } \{1,2,\ldots,n\}}} \underbrace{\prod_{(i,j) \text{ is an arc of } D} \begin{cases} x_j, & \text{if } i < j; \\ -x_j, & \text{if } i \geq j \end{cases}}_{\substack{= (-1)^{(\text{\# of red arcs of } D)} \prod_{j=1}^{n} x_j^{\deg^- j} \\ \text{(where } \deg^- j \text{ means the indegree of } j \text{ in } D\text{,} \\ \text{and where the ``red arcs'' are the arcs } ij \text{ with } i > j\text{)}}}$$

$$= \sum_{\substack{D \text{ is a tournament} \\ \text{with vertex set } \{1,2,\ldots,n\}}} (-1)^{(\text{\# of red arcs of } D)} \prod_{j=1}^{n} x_j^{\deg^- j}.$$

We shall refer to this sum as 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 σ by $\operatorname{sign} \sigma$, then we have

$$\det V = \det \left(V^T \right) = \sum_{\sigma \in S_n} \operatorname{sign} \sigma \cdot \prod_{j=1}^{n} x_j^{\sigma(j)-1}$$

(by the definition of a determinant). We shall refer to this sum as the "small sum".

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

1. Each addend of the small sum is an addend of the big sum. Indeed, for each permutation $\sigma \in S_n$, there is a certain tournament T_σ that has

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

   Can you find this T_σ ?

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

   The basic 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., we reverse each of its arcs), the indegrees of all vertices are preserved, but the sign $(-1)^{(\text{\# of red arcs of } D)}$ is flipped (since three arcs change their orientation).

   This suffices to show that for each addend that appears in the big sum but not in the small sum, there is another addend with the same magnitude but with opposite sign. Unfortunately, this in itself does not suffice to ensure that all these addends cancel out; for example, the sum $1+1+1+(-1)$ has the same property but does not equal 0. We need to show that the # of addends with positive sign (i.e., with $(-1)^{(\text{\# of red arcs of } D)} = 1$) and a given magnitude equals the # of addends with negative sign (i.e., with $(-1)^{(\text{\# of red arcs of } D)} = -1$) and the same magnitude.

One way to achieve this would be by constructing a bijection (aka "perfect matching") between the "positive" and the "negative" addends. This is tricky here: We would have to decide **which** 3-cycle to reverse (as there are usually many of them), and this has to be done in a bijective way (so that two "positive" addends don't get assigned the same "negative" partner).

A less direct, but easier way is the following: Fix a positive integer k, and consider only the tournaments with exactly k many 3-cycles. For each such tournament, we can reverse any of its k many 3-cycles. It can be shown (nice exercise!) that reversing the arcs of a 3-cycle does not change the # of all 3-cycles; thus, we don't accidentally change our k in the process. Thus, we find a "k-to-k" correspondence between the "positive" addends of a given magnitude and the "negative" addends of the same magnitude. As one can easily see, this entails that the former and the latter are equinumerous, and thus really cancel out. The addends that remain are exactly those in the small sum.

As already mentioned, this is only a rough summary of the proof; the details can be found in [17s-lec8].

4.11. Exercises on tournaments

There is, of course, much more to say about tournaments. See [Moon13] for a selection of topics. Let us merely hint at some possible directions by giving a few exercises.

The next three exercises use the notion of a "3-cycle":

Definition 4.11.1. A **3-cycle** in a tournament $D = (V, A)$ means a triple (u, v, w) of vertices in V such that all three pairs (u, v), (v, w) and (w, u) belong to A.

For example, the tournament shown in Example 4.10.4 has the nine different 3-cycles

$$\begin{array}{llll} (1,4,3), & (1,5,3), & (2,5,3), & (3,1,4), \\ (3,1,5), & (3,2,5), & (4,3,1), & (5,3,1), \\ (5,3,2). & & & \end{array}$$

(Yes, we are counting a 3-cycle (u, v, w) as being distinct from (v, w, u) and (w, u, v).)

Exercise 4.12. Let $D = (V, A)$ be a tournament. Set $n = |V|$ and $m = \sum_{v \in V} \binom{\deg^{-}(v)}{2}$.

(a) Show that $m = \sum_{v \in V} \binom{\deg^{+}(v)}{2}$.

(b) Show that the number of 3-cycles in D is $3\left(\binom{n}{3} - m\right)$.

[**Solution:** This is Exercise 5 on homework set #2 from my Spring 2017 course; see the course page for solutions.]

The next exercise uses the notation $\deg_D^{-} v$ for the indegree of a vertex v in a digraph D. (We usually denote this by $\deg^{-} v$, but sometimes it is important to stress the dependence on D, since v can be a vertex of two different digraphs.)

Exercise 4.13. If a tournament D has a 3-cycle (u, v, w), then we can define a new tournament $D'_{u,v,w}$ as follows: The vertices of $D'_{u,v,w}$ shall be the same as those of D. The arcs of $D'_{u,v,w}$ shall be the same as those of D, except that the three arcs (u, v), (v, w) and (w, u) are replaced by the three new arcs (v, u), (w, v) and (u, w). (Visually speaking, $D'_{u,v,w}$ is obtained from D by turning the arrows on the arcs (u, v), (v, w) and (w, u) around.) We say that the new tournament $D'_{u,v,w}$ is obtained from the old tournament D by a **3-cycle reversal operation**.

Now, let V be a finite set, and let E and F be two tournaments with vertex set V. Prove that F can be obtained from E by a sequence of 3-cycle reversal operations if and only if each $v \in V$ satisfies $\deg_E^{-}(v) = \deg_F^{-}(v)$. (Note that a sequence may be empty, which allows handling the case $E = F$ even if E has no 3-cycles to reverse.)

[**Solution:** This is Exercise 6 on homework set #2 from my Spring 2017 course; see the course page for solutions.]

Exercise 4.14. A tournament $D = (V, A)$ is called **transitive** if it has no 3-cycles.

If a tournament $D = (V, A)$ has three distinct vertices u, v and w satisfying $(u, v) \in A$ and $(v, w) \in A$, then we can define a new tournament $D''_{u,v,w}$ as follows: The vertices of $D''_{u,v,w}$ shall be the same as those of D. The arcs of $D''_{u,v,w}$ shall be the same as those of D, except that the two arcs (u, v) and (v, w) are replaced by the two new arcs (v, u) and (w, v). We say that the new tournament $D''_{u,v,w}$ is obtained from the old tournament D by a **2-path reversal operation**.

Let D be any tournament. Prove that there is a sequence of 2-path reversal operations that transforms D into a transitive tournament.

[**Solution:** This is Exercise 7 on homework set #2 from my Spring 2017 course; see the course page for solutions.]

5. Trees and arborescences

Trees are particularly nice graphs. Among other things, they can be characterized as

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

Arborescences are their closest analogue for digraphs.

In this chapter, we will discuss the theory of trees and some of their applications. Further applications are usually covered in courses in theoretical computer science, but their notion of a tree is somewhat different from ours.

5.1. Some general properties of components and cycles

5.1.1. Backtrack-free walks revisited

Before we start with trees, let us recall and prove some more facts about general multigraphs. Recall the notion of a "backtrack-free walk" that already had a brief appearance in the proof of Theorem 2.10.7:

Definition 5.1.1. Let G be a multigraph. A **backtrack-free walk** of G means a walk $\mathbf{w}$ such that no two adjacent edges of $\mathbf{w}$ are identical.

Here are a few properties of this notion:

Proposition 5.1.2. 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.

Proof. We have already proved this for simple graphs (in Proposition 2.10.4). More or less the same argument works for multigraphs. ("More or less" because the definition of a cycle in a multigraph is slightly different from that in a simple graph; but the proof is easy to adapt.) ☐

Theorem 5.1.3. Let G be a multigraph. Let u and v be two vertices of G. Assume that there are two distinct backtrack-free walks from u to v in G. Then, G has a cycle.

Proof. We have already proved this for simple graphs (Claim 1 in the proof of Theorem 2.10.7). More or less the same argument works for multigraphs. ☐

5.1.2. Counting components

Next, we shall derive a few properties of the number of components of a graph. Again, we have already done most of the hard work, and we can now derive corollaries. First, we give this number a name:

Definition 5.1.4. Let G be a multigraph. Then, $\operatorname{conn} G$ means the number of components of G. (Some authors also call this number $b_0(G)$. This notation comes from algebraic topology, where it stands for the 0-th Betti number. This makes sense, because we can regard a multigraph G as a topological space. But we won't need this.)

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

Let us next recall Definition 3.3.17 and Theorem 3.3.18 (which is an analogue of Theorem 2.12.2 and can be proved in more or less the same way). As a consequence of the latter theorem, we obtain the following:

Corollary 5.1.5. 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 $\operatorname{conn}(G \setminus e) = \operatorname{conn} G$.
- **(b)** If e appears in no cycle of G, then $\operatorname{conn}(G \setminus e) = \operatorname{conn} G + 1$.
- **(c)** In either case, we have $\operatorname{conn}(G \setminus e) \leq \operatorname{conn} G + 1$.

Proof. Part **(a)** follows from Theorem 3.3.18 **(a)**. Part **(b)** follows from Theorem 3.3.18 **(b)**. Part **(c)** follows by combining parts **(a)** and **(b)**. □

Corollary 5.1.6. Let $G = (V, E, \varphi)$ be a multigraph. Then, $\operatorname{conn} G \geq |V| - |E|$.

Proof. We induct on $|E|$:

Base case: If $|E| = 0$, then $\operatorname{conn} G = |V|$ (since $|E| = 0$ means that the graph G has no edges, and thus no two distinct vertices are path-connected); but this rewrites as $\operatorname{conn} G = |V| - |E|$ (since $|E| = 0$). Thus, Corollary 5.1.6 is proved for $|E| = 0$.

Induction step: Let $k \in \mathbb{N}$. Assume (as the induction hypothesis) that Corollary 5.1.6 holds for $|E| = k$. We must now show that it also holds for $|E| = k + 1$.

So let us consider a multigraph $G = (V, E, \varphi)$ with $|E| = k + 1$. Thus, $|E| - 1 = k$. Pick any edge $e \in E$ (such an edge exists, since $|E| = k + 1 \geq 1 > 0$). Then, the multigraph $G \setminus e$ has edge set $E \setminus \{e\}$ and therefore has $|E \setminus \{e\}| = |E| - 1 = k$ many edges. Hence, by the induction hypothesis, we have

$$\operatorname{conn}(G \setminus e) \geq |V| - |E \setminus \{e\}|$$

(since $G \setminus e$ is a multigraph with vertex set V and edge set $E \setminus \{e\}$). However, Corollary 5.1.5 **(c)** yields $\operatorname{conn}(G \setminus e) \leq \operatorname{conn} G + 1$. Thus,

$$\operatorname{conn} G \geq \underbrace{\operatorname{conn}(G \setminus e)}_{\geq |V| - |E \setminus \{e\}|} - 1 \geq |V| - \underbrace{|E \setminus \{e\}|}_{=|E|-1} - 1 = |V| - (|E| - 1) - 1 = |V| - |E|.$$

This completes the induction step. Thus, Corollary 5.1.6 is proven. □

Corollary 5.1.7. Let $G = (V, E, \varphi)$ be a multigraph that has no cycles. Then, $\operatorname{conn} G = |V| - |E|$.

Proof. Replay the proof of Corollary 5.1.6, with just a few changes: Instead of applying Corollary 5.1.5 **(c)**, apply Corollary 5.1.5 **(b)** (this is allowed because G has no cycles and thus e appears in no cycle of G). The induction hypothesis can be used because when G has no cycles, $G \setminus e$ has no cycles either. All $\leq$ and $\geq$ signs in the above proof now can be replaced by $=$ signs (since Corollary 5.1.5 **(b)** claims an equality, not an inequality). The result is therefore $\operatorname{conn} G = |V| - |E|$. □

Corollary 5.1.8. Let $G = (V, E, \varphi)$ be a multigraph that has at least one cycle. Then, $\operatorname{conn} G \geq |V| - |E| + 1$.

Proof. Pick an edge $e \in E$ that belongs to some cycle (such an edge exists, since G has at least one cycle). Then, Corollary 5.1.5 **(a)** yields $\operatorname{conn}(G \setminus e) = \operatorname{conn} G$. However, Corollary 5.1.6 (applied to $G \setminus e$ and $E \setminus \{e\}$ instead of G and E) yields

$$\operatorname{conn}(G \setminus e) \geq |V| - \underbrace{|E \setminus \{e\}|}_{=|E|-1} = |V| - (|E| - 1) = |V| - |E| + 1.$$

Since $\operatorname{conn}(G \setminus e) = \operatorname{conn} G$, this rewrites as $\operatorname{conn} G \geq |V| - |E| + 1$. □

We summarize what we have proved into one convenient theorem:

Theorem 5.1.9. 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.

Proof. **(a)** This is Corollary 5.1.6.

(b) $\Longleftarrow$: This is Corollary 5.1.7.

$\Longrightarrow$: Assume that $\operatorname{conn} G = |V| - |E|$. If G had any cycles, then Corollary 5.1.8 would yield $\operatorname{conn} G \geq |V| - |E| + 1 > |V| - |E|$, which would contradict $\operatorname{conn} G = |V| - |E|$. So G has no cycles. This proves the "$\Longrightarrow$" direction of Theorem 5.1.9. □

Remark 5.1.10. Let $G = (V, E, \varphi)$ be a multigraph. Does the number

$$\operatorname{conn} G - (|V| - |E|)$$

have anything to do with how many cycles G has? We know that it is 0 if G has no cycles. More generally, could it just be the number of cycles of G ? (Let's say we count reversals and cyclic rotations of a cycle as being the same cycle.)

Unfortunately, the answer is still no. For example, a complete graph K_n has many more than $1 - \left(n - \binom{n}{2}\right)$ many cycles. However, there is still some subtler connection. The number $\operatorname{conn} G - (|V| - |E|)$ is known as the **circuit rank** or the **cyclomatic number** of G, and is the dimension of a certain vector space that, in some way, consists of cycles.

5.2. Forests and trees

5.2.1. Definitions

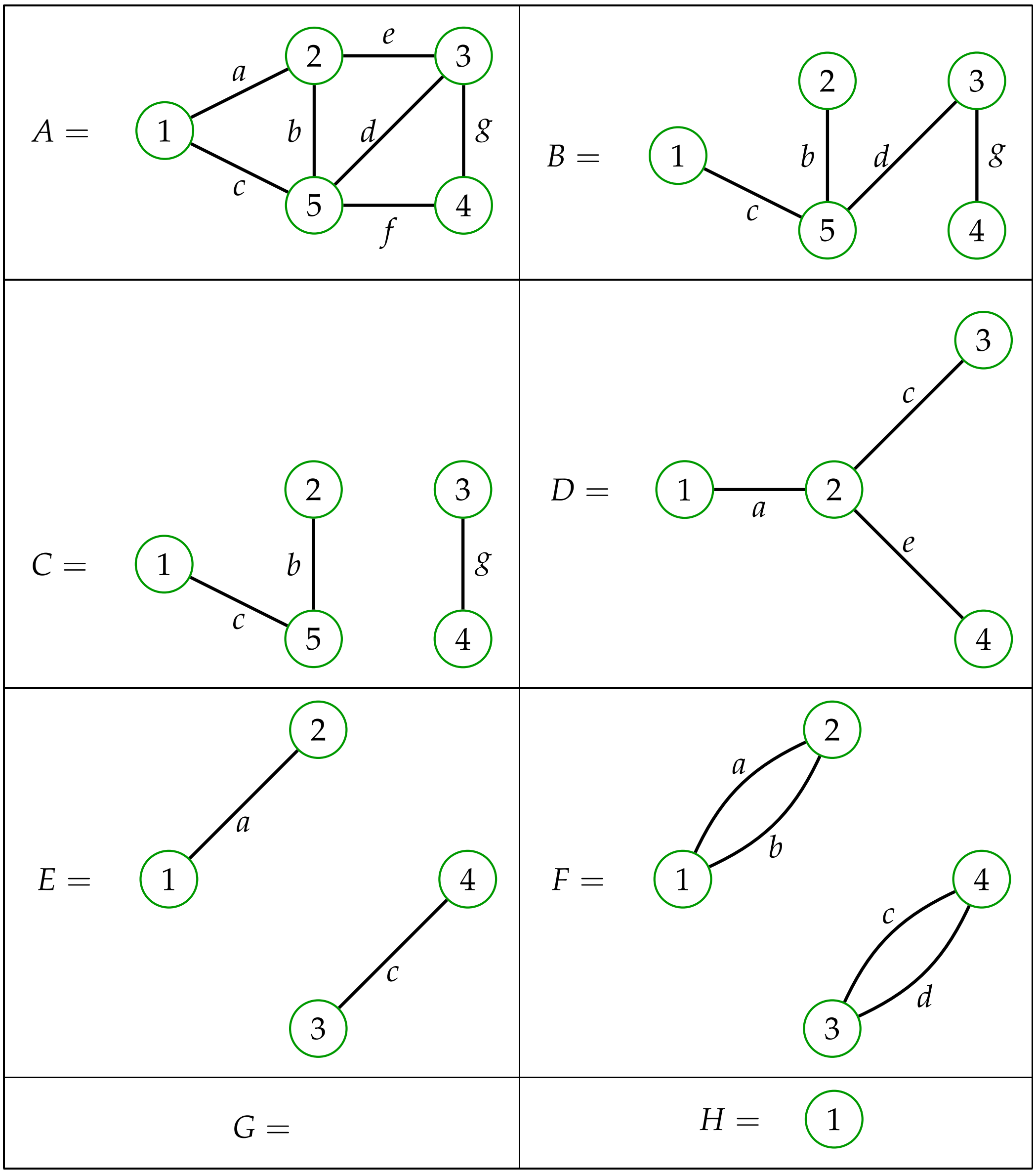
We now introduce two of the heroes of this chapter:

Definition 5.2.1. A **forest** is a multigraph with no cycles.

(In particular, a forest therefore cannot contain two distinct parallel edges. It also cannot contain loops.)

Definition 5.2.2. A **tree** is a connected forest.

Example 5.2.3. Consider the following multigraphs:



(Yes, G is an empty graph with no vertices.) Which of them are forests, and which are trees?

- The graph A is not a forest, since it has a cycle (actually, several cycles). Thus, A is not a tree either.
- The graph B is a tree.
- The graph C is a forest, but not a tree, since it is not connected.
- The graph D is a tree.
- The graph E is a forest, but not a tree.

- The graph F is not a forest, since it has cycles.
- The graph G (which has no vertices and no edges) is a forest, but not a tree, since it is not connected (recall: a graph is connected if it has 1 component; but G has 0 components).
- The graph H is a tree.

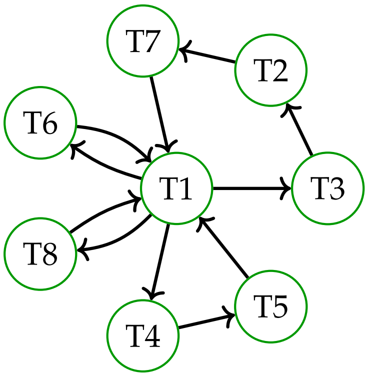
5.2.2. The tree equivalence theorem

Trees can be described in many ways:

Theorem 5.2.4 (The 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 $u \in V$ and $v \in V$, there is a **unique** path from u to v.
- **Statement T3:** We have $V \neq \varnothing$, and for each $u \in V$ and $v \in V$, 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 (i.e., non-connected) graph.
- **Statement T8:** The multigraph G is a forest, and we have $|E| \geq |V| - 1$ and $V \neq \varnothing$.

Proof. We shall prove the following implications:



.

In this digraph, an arc from Ti to Tj stands for the implication Ti $\Longrightarrow$Tj. Since this digraph is strongly connected (i.e., you can travel from Statement Ti to Statement Tj along its arcs for any i, j), this will prove the theorem. So let us prove the implications.

Proof of T1$\Longrightarrow$T3: Assume that Statement T1 holds. Thus, G is a tree. Therefore, G is connected, so that $V \neq \varnothing$. We must prove that for each $u \in V$ and $v \in V$, there is a **unique** backtrack-free walk from u to v. The existence of such a walk is clear (since G is connected, so there is a path from u to v). Thus, we only need to show that it is unique. But this is easy: If there were two distinct backtrack-free walks from u to v (for some $u \in V$ and $v \in V$), then Theorem 5.1.3 would show that G has a cycle, and thus G could not be a forest, let alone a tree. Thus, the backtrack-free walk from u to v is unique. So we have proved Statement T3. The implication T1$\Longrightarrow$T3 is thus proved.

Proof of T3$\Longrightarrow$T2: Assume that Statement T3 holds. We must prove that Statement T2 holds. First, G has no loops, because if there was a loop e with endpoint u, then the two walks (u) and (u, e, u) would be two distinct backtrack-free walks from u to u. It remains to prove that for each each $u \in V$ and $v \in V$, there is a **unique** path from u to v. However, the existence of a walk from u to v always implies the existence of a path from u to v (by Corollary 3.3.10). Moreover, the uniqueness of a backtrack-free walk from u to v implies the uniqueness of a path from u to v (since any path is a backtrack-free walk). Thus, Statement T2 follows from Statement T3.

Proof of T2$\Longrightarrow$T7: Assume that Statement T2 holds. Then, G is connected. Now, let us remove any edge e from G. Let u and v be the endpoints of e. Then, $u \neq v$ (since G has no loops). There cannot be a path from u to v in the graph $G \setminus e$ (because if there was such a path, then it would also be a path from u to v in the graph G, and this path would be distinct from the path (u, e, v); thus, the graph G would have at least two paths from u to v; but this would contradict the uniqueness part of Statement T2). Hence, the graph $G \setminus e$ is disconnected. So we have shown that G is connected, but removing any edge from G yields a disconnected graph. In other words, Statement T7 holds.

Proof of T7$\Longrightarrow$T1: Assume that Statement T7 holds. We must show that G is a tree. Since G is connected (by Statement T7), it suffices to show that G is a forest, i.e., that G has no cycles. However, if G had any cycle, then we could pick any edge e of this cycle, and then we would know that $G \setminus e$ is still connected (since Corollary 5.1.5 **(a)** would yield $\operatorname{conn}(G \setminus e) = \operatorname{conn} G = 1$), and this would contradict Statement T7. Thus, G has no cycles, hence is a forest. This proves Statement T1.

Proof of T1$\Longrightarrow$T6: Assume that Statement T1 holds. Thus, G is a tree. We must show that adding any new edge to G creates a cycle (since all other parts of Statement T6 are clear).

Indeed, let us add a new edge f to G. Let u and v be the endpoints of f. The graph G is connected, so there is already a path from u to v in G. Combining this path with the edge f, we obtain a cycle. Thus, the graph obtained from G by adding the new edge f has a cycle. This completes our proof that Statement T6 holds.

Proof of T6$\Longrightarrow$T1: Assume that Statement T6 holds. Thus, G is a forest. We must only show that G is connected.

Assume the contrary. Thus, there exist two vertices u and v of G that are not path-connected in G. Hence, adding a new edge f with endpoints u and v to the graph G cannot create a new cycle (because any such cycle would have to contain f (otherwise, it would already be a cycle of G, but G has no cycles), and then we could remove f from it to obtain a path from u to v in G; but such a path cannot exist, since u and v are not path-connected in G). This contradicts Statement T6.

So we have shown that G is connected, and thus G is a tree. This proves Statement T1.

Proof of T1$\Longrightarrow$T8: Assume that Statement T1 holds. So G is a tree. Clearly, G is then a forest. We must show that $|E| \geq |V| - 1$.

Theorem 5.1.9 **(a)** yields $\operatorname{conn} G \geq |V| - |E|$. But we have $\operatorname{conn} G = 1$ because G is connected. Thus, $1 = \operatorname{conn} G \geq |V| - |E|$. In other words, $|E| \geq |V| - 1$. This proves Statement T8.

Proof of T8$\Longrightarrow$T1: Assume that Statement T8 holds. Thus, G is a forest. We must only show that G is connected. However, G is a forest, and thus has no cycles. Hence, Theorem 5.1.9 **(b)** yields $\operatorname{conn} G = |V| - |E| \leq 1$ (since Statement 8 yields $|E| \geq |V| - 1$). On the other hand, $\operatorname{conn} G \geq 1$ (since $V \neq \varnothing$). Combining these two inequalities, we obtain $\operatorname{conn} G = 1$. In other words, G is connected. This yields Statement T1 (since G is a forest).

Proof of T1$\Longrightarrow$T4: Assume that Statement T1 holds. Then, G is a tree, hence a connected forest. Therefore, G has no cycles (by the definition of a forest). Theorem 5.1.9 **(b)** therefore yields $\operatorname{conn} G = |V| - |E|$. Thus, $|V| - |E| = \operatorname{conn} G = 1$ (since G is connected), so that $|E| = |V| - 1$. Thus, Statement T4 is proved.

Proof of T4$\Longrightarrow$T5: The implication T4$\Longrightarrow$T5 is obvious.

Proof of T5$\Longrightarrow$T1: Assume that Statement T5 holds. Thus, the multigraph G is connected, and we have $|E| < |V|$. Thus, $|E| \leq |V| - 1$. In other words, $1 \leq |V| - |E|$. Since G is connected, we have $\operatorname{conn} G = 1 \leq |V| - |E|$. However, Theorem 5.1.9 **(a)** yields $\operatorname{conn} G \geq |V| - |E|$. Combining these two inequalities, we obtain $\operatorname{conn} G = |V| - |E|$. Thus, Theorem 5.1.9 **(b)** shows that G has no cycles. In other words, G is a forest. Hence, G is a tree (since G is connected). This proves Statement T1.

We have now proved all necessary implications to conclude that all eight statements T1, T2, ..., T8 are equivalent. Theorem 5.2.4 is thus proved. □

We also observe the following connection between trees and forests:

Proposition 5.2.5. 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. $\Longrightarrow$: Assume that G is a forest. Thus, G has no cycles. Hence, the induced subgraphs $G[C_1], G[C_2], \ldots, G[C_k]$ have no cycles either (since a cycle in any of them would be a cycle of G); in other words, they are forests. But they are furthermore connected (since the induced subgraph on a component is always connected). Hence, they are connected forests, i.e., trees.

$\Longleftarrow$: Assume that the induced subgraphs $G[C_1], G[C_2], \ldots, G[C_k]$ are trees. Hence, none of them has a cycle. Thus, G has no cycles either (since a cycle of G would have to be fully contained in one of these induced subgraphs[31]). In other words, G is a forest. □

5.2.3. Summary

Let us briefly summarize some properties of trees:

If $T = (V, E, \varphi)$ is a tree, then...

- T is a connected forest. (This is how trees were defined.) Thus, T has no cycles. (This is how forests were defined.)
- we have $|E| = |V| - 1$. (This follows from the implication T1$\Longrightarrow$T4 in Theorem 5.2.4.)
- adding any new edge to T creates a cycle. (This follows from the implication T1$\Longrightarrow$T6 in Theorem 5.2.4.)
- removing any edge from T yields a disconnected (i.e., non-connected) graph. (This follows from the implication T1$\Longrightarrow$T7 in Theorem 5.2.4.)

[31] Indeed, if it wasn't, then it would contain vertices from different components. But this is impossible, since there are no walks between vertices in different components.

- for each $u \in V$ and $v \in V$, there is a **unique** backtrack-free walk from u to v. (This follows from the implication T1$\Longrightarrow$T3 in Theorem 5.2.4.) Moreover, this backtrack-free walk is a path (since any walk from u to v contains a path from u to v).

Remark 5.2.6. Computer scientists use some notions of "trees" that are similar to ours, but not quite the same. In particular, their trees often have **roots** (i.e., one vertex is chosen to be called "the root" of the tree), which leads to a parent/child relationship on each edge (namely: the endpoint closer to the root is called the "parent" of the endpoint further away from the root). Often, they also impose a total order on the children of each given vertex. With these extra data, a tree can be used for addressing objects, since each vertex has a unique "path description" from the root leading to it (e.g., "the second child of the fourth child of the root"). But this all is going too far afield for us here; we are mainly interested in trees as graphs, and won't impose any extra structure unless we need it for something.

Exercise 5.1. Let G be a multigraph that has no loops. Assume that there exists a vertex u of G such that

for each vertex v of G, there is a **unique** path from u to v in G.

Prove that G is a tree.

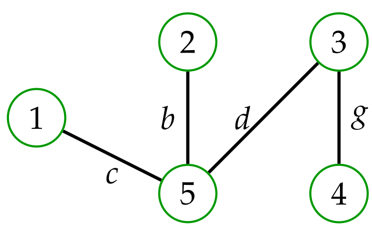
[**Remark:** Pay attention to the quantifiers used here: $\exists u \forall v$. This differs from the $\forall u \forall v$ in Statement T2 of the tree equivalence theorem (Theorem 5.2.4).]

5.3. Leaves

Continuing with our faux-botanical terminology, we define leaves in a tree:

Definition 5.3.1. Let T be a tree. A vertex of T is said to be a **leaf** if its degree is 1.

For example, the tree

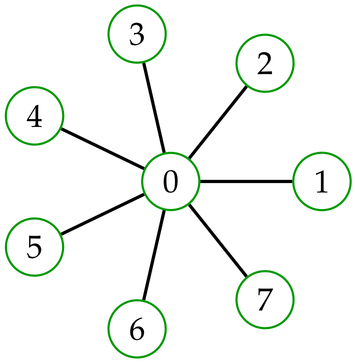


has three leaves: 1, 2 and 4.

How to find a tree with as many leaves as possible (for a given number of vertices)? For any $n \geq 3$, the simple graph

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

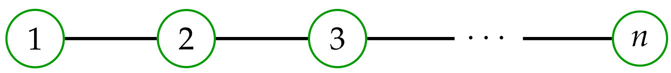
is a tree (when considered as a multigraph), and has $n-1$ leaves (namely, all of $1, 2, \ldots, n-1$). This tree is called an **n-star graph**, as it looks as follows:



(for $n = 8$).

It is easy to see that no tree with $n \geq 3$ vertices can have more than $n-1$ leaves, so the n-star graph is optimal in this sense. Note that for $n = 2$, the n-star graph has 2 leaves, not 1.

How to find a tree with as few leaves as possible? For any $n \geq 2$, the n-path graph

$P_n =$ 

is a tree with only 2 leaves (viz., the vertices 1 and n). Can we find a tree with fewer leaves? For $n = 1$, yes, because the 1-path graph P_1 (this is simply the graph with 1 vertex and no edges) has no leaves at all. However, for $n \geq 2$, the n-path graph is the best we can do:

Theorem 5.3.2. Let T be a tree with at least 2 vertices. Then:

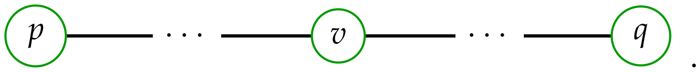
(a) The tree T has at least 2 leaves.

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

Note that I'm saying "the path" rather than "a path" here. This is allowed, because in a tree, for any two vertices p and q, there is a **unique** path from p to q. This follows from Statement T2 in the tree equivalence theorem (Theorem 5.2.4).

Proof of Theorem 5.3.2. **(b)** We apply a variant of the "longest path trick": Among all paths that contain the vertex v, let $\mathbf{w}$ be a longest one. Let p be the starting point of $\mathbf{w}$, and let q be the ending point of $\mathbf{w}$. We shall show that p and q are two distinct leaves.

[Here is a picture of $\mathbf{w}$, for what it's worth:



Of course, the tree T can have other edges as well, not just those of $\mathbf{w}$.]

First, we observe that T is connected (since T is a tree), and has at least one vertex u distinct from v (since T has at least 2 vertices). Hence, T has a path $\mathbf{r}$ that connects v to u. This path $\mathbf{r}$ must contain at least one edge (since $u \neq v$). Thus, we have found a path $\mathbf{r}$ of T that contains v and contains at least one edge. Hence, the path $\mathbf{w}$ must contain at least one edge as well (since $\mathbf{w}$ is a longest path that contains v, and thus cannot be shorter than $\mathbf{r}$). Since $\mathbf{w}$ is a path from p to q, we thus conclude that $p \neq q$ (because if a path contains at least one edge, then its starting point is distinct from its ending point).

Now, assume (for the sake of contradiction) that p is not a leaf. Then, $\deg p \neq 1$. The path $\mathbf{w}$ already contains one edge that contains p (namely, the first edge of $\mathbf{w}$). Since $\deg p \neq 1$, there must be another edge f of T that contains $\mathbf{w}$. Consider this f. Let p' be its endpoint distinct from p (if f is a loop, then we set $p' = p$). Appending this edge f (and its endpoint) to the beginning of the path $\mathbf{w}$, we obtain a backtrack-free walk

$$\left(p', f, \underbrace{p, \ldots, v, \ldots, q}_{\text{This is } \mathbf{w}}\right)$$

(this is backtrack-free since f is not the first edge of $\mathbf{w}$). According to Proposition 5.1.2, this backtrack-free walk either is a path or contains a cycle. Since T has no cycle (because T is a forest), we thus conclude that this backtrack-free walk is a path. It is furthermore a path that contains v and is longer than $\mathbf{w}$ (longer by 1, in fact). But this contradicts the fact that $\mathbf{w}$ is a longest path that contains v. This contradiction shows that our assumption (that p is not a leaf) was wrong.

Hence, p is a leaf. A similar argument shows that q is a leaf (here, we need to append the new edge at the end of $\mathbf{w}$ rather than at the beginning). Thus, p and q are two distinct leaves of T (distinct because $p \neq q$) such that v lies on the path from p to q (since v lies on the path $\mathbf{w}$, which is a path from p to q). This proves Theorem 5.3.2 **(b)**.

(a) Pick any vertex v of T. Then, Theorem 5.3.2 **(b)** shows that there exist two distinct leaves p and q of T such that v lies on the path from p to q. Thus, in particular, there exist two distinct leaves p and q of T. In other words, T has at least two leaves. This proves Theorem 5.3.2 **(a)**.

[*Remark:* Another way to prove part **(a)** is to write the tree T as $T = (V, E, \varphi)$, and recall the handshake lemma, which yields

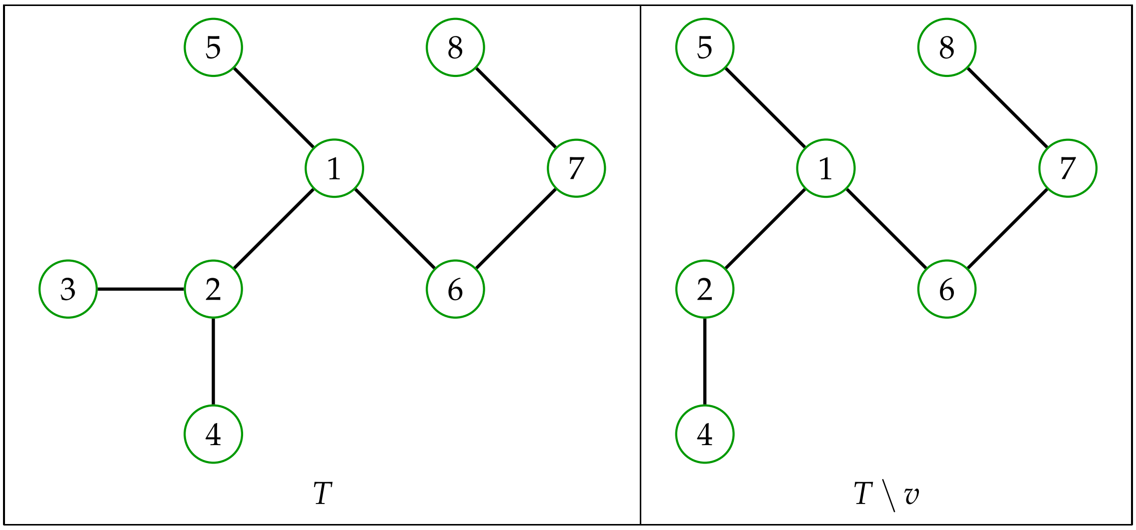
$$\sum_{v \in V} \deg v = 2 \cdot |E| = 2 \cdot (|V| - 1) \qquad \text{(since } |E| = |V| - 1 \text{ in a tree)}$$
$$= 2 \cdot |V| - 2.$$

Since each $v \in V$ satisfies $\deg v \geq 1$ (why?), this equality entails that at least two vertices $v \in V$ must satisfy $\deg v \leq 1$ (since otherwise, the sum $\sum_{v \in V} \deg v$ would be $\geq 2 \cdot |V| - 1$), and therefore these two vertices are leaves.] $\square$

Leaves are particularly helpful for performing induction on trees. The formal reason for this is the following theorem:

Theorem 5.3.3 (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 that contain v (note that there is only one such edge, since v is a leaf). Then, $T \setminus v$ is again a tree.

Here is an example of a tree T and of the smaller tree $T \setminus v$ obtained by removing a leaf v (namely, $v = 3$):



Proof of Theorem 5.3.3. Write T as $T = (V, E, \varphi)$. Thus, $T \setminus v$ is the induced subgraph $T[V \setminus \{v\}]$.

The graph T is a tree, thus a forest; hence, it has no cycles. Thus, the graph $T \setminus v$ has no cycles either. Hence, it is a forest.

Furthermore, this forest $T \setminus v$ has at least 1 vertex (since T has at least 2 vertices).

We shall now show that any two vertices p and q of $T \setminus v$ are path-connected in $T \setminus v$.

Indeed, let p and q be two vertices of $T \setminus v$. Then, p and q are path-connected in T (since T is connected). Hence, there exists a path $\mathbf{w}$ from p to q in T. Consider this path $\mathbf{w}$. Note that v is neither the starting point nor the ending point of this path $\mathbf{w}$ (since p and q are vertices of $T \setminus v$, and thus distinct from v). Hence, if v was a vertex of $\mathbf{w}$, then $\mathbf{w}$ would contain **two distinct** edges that contain v (namely, the edge just before v and the edge just after v). But this is impossible, since there is only one edge available that contains v (because v is a leaf). Thus, v cannot be a vertex of $\mathbf{w}$. Hence, the path $\mathbf{w}$ does not use the vertex v, and thus is a path in the graph $T \setminus v$ as well. So the vertices p and q are path-connected in $T \setminus v$.

We have now shown that any two vertices p and q of $T \setminus v$ are path-connected in $T \setminus v$. This shows that $T \setminus v$ is connected (since $T \setminus v$ has at least 1 vertex). Hence, $T \setminus v$ is a tree (since $T \setminus v$ is a forest). ☐

Theorem 5.3.3 has a converse as well:

Theorem 5.3.4. 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. (Here, $G \setminus v$ means the multigraph obtained from G by removing the vertex v and all edges that contain v.) Then, G is a tree.

Proof. Left to the reader. (The main step is to show that a cycle of G cannot contain v.) ☐

Theorem 5.3.3 helps prove many properties of trees by induction on the number of vertices. In the induction step, remove a leaf v and apply the induction hypothesis to $T \setminus v$.

The following exercise is essentially a generalization of Theorem 5.3.2 **(a)**:

Exercise 5.2. Let T be a tree. Let w be any vertex of T. Prove that T has at least $\deg w$ many leaves.

Exercise 5.3. A dominating set of a multigraph G is defined to be a dominating set of its underlying simple graph G^{simp}.

Let G be a forest. Prove that

$$\sum_{D \text{ is a dominating set of } G} (-1)^{|D|} = \pm 1.$$

Exercise 5.4. Let T be a tree having more than 1 vertex. Let L be the set of leaves of T. Prove that it is possible to add $|L| - 1$ new edges to T in such a way that the resulting multigraph has a Hamiltonian cycle.[**Solution:** This is Exercise 4 on homework set #3 from my Spring 2017 course; see the course page for solutions.]

5.4. Spanning trees

5.4.1. Spanning subgraphs

We now proceed to a crucial application of trees. First we define a concept that makes sense for any multigraphs:

Definition 5.4.1. 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.

In other words, it means a multigraph obtained from G by removing some edges, but leaving all vertices undisturbed.

Compare this to the notion of an induced subgraph:

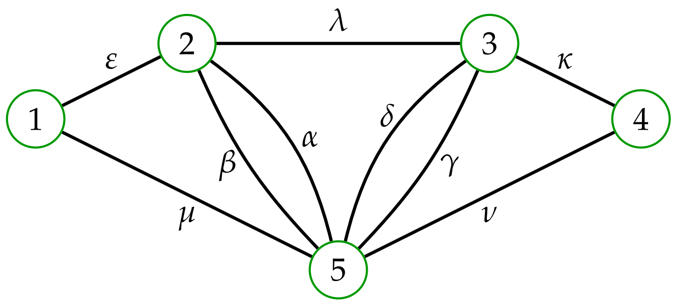
- To build an **induced** subgraph, we throw away some vertices but keep all the edges that we can keep. (As usual in mathematics, the words "some vertices" include "no vertices" and "all vertices".)
- In contrast, to build a **spanning** subgraph, we keep all vertices but throw away some edges.

5.4.2. Spanning trees

Spanning subgraphs are particularly useful when they are trees:

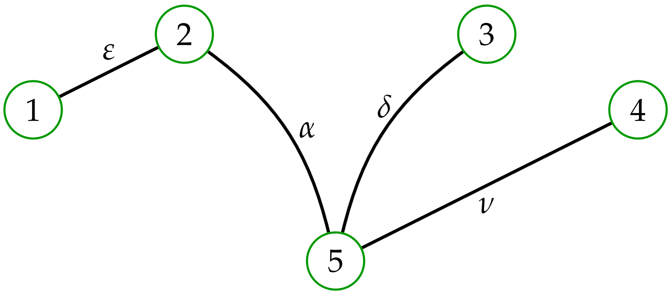
Definition 5.4.2. A **spanning tree** of a multigraph G means a spanning subgraph of G that is a tree.

Example 5.4.3. Let G be the following multigraph:



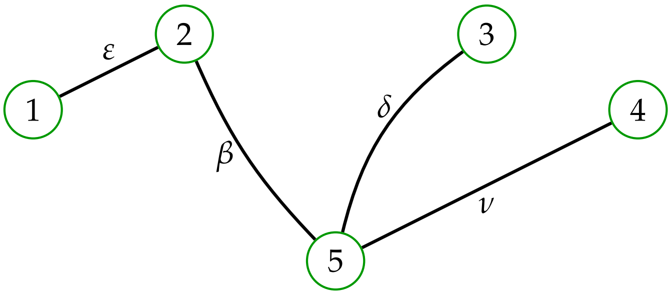
.

Here is a spanning tree of G:



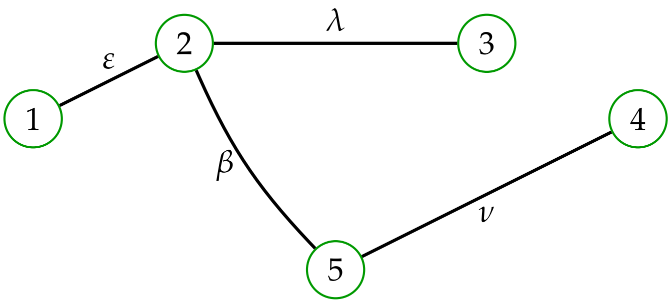
.

Here is another:



.

(Yes, this is a different one, because $\alpha \neq \beta$.) And here is yet another spanning tree of G:



.

Example 5.4.4. Let n be a positive integer. Consider the cycle graph C_n. (We defined this graph C_n in Definition 2.6.3 for all $n \geq 2$, but we later redefined C_2 and defined C_1 in Definition 3.3.5. Here, we are using the latter modified definition.)

The graph C_n has exactly n spanning trees. Indeed, any graph obtained from C_n by removing a single edge is a spanning tree of C_n.

Proof. A tree with n vertices must have exactly $n-1$ edges (by the implication T1$\Longrightarrow$T4 in Theorem 5.2.4). Thus, a spanning subgraph of C_n can be a tree only if it has $n-1$ edges, i.e., only if it is obtained from C_n by removing a single edge (since C_n has n edges in total). Thus, C_n has at most n spanning trees (since C_n has n edges that can be removed). It remains to check that any subgraph obtained from C_n by removing a single edge is indeed a spanning tree. But this is easy, since all such subgraphs are isomorphic to the path graph P_n. This proves Example 5.4.4. □

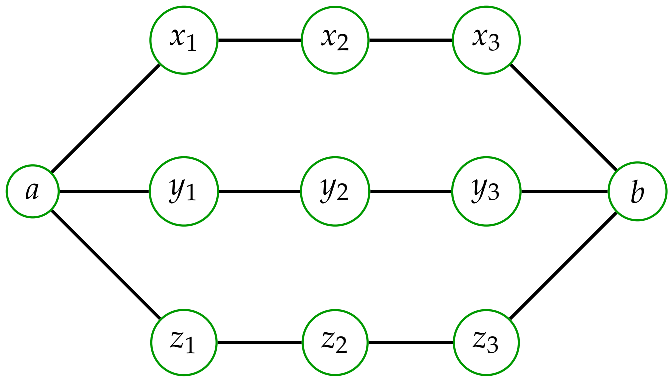
Exercise 5.5. Fix $m \geq 1$. Let G be the simple graph with $3m + 2$ vertices

$$a, b, x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m, z_1, z_2, \ldots, z_m$$

and the following $3m + 3$ edges:

$$\begin{aligned} & ax_1, ay_1, az_1, \\ & x_i x_{i+1}, y_i y_{i+1}, z_i z_{i+1} \qquad \text{for all } i \in \{1, 2, \ldots, m-1\}, \\ & x_m b, y_m b, z_m b. \end{aligned}$$

(Thus, the graph consists of two vertices a and b connected by three paths, each of length $m + 1$, with no overlaps between the paths except for their starting and ending points. Here is a picture for $m = 3$:



) Compute the number of spanning trees of G.

[To argue why your number is correct, a sketch of the argument in 1-2 sentences should be enough; a fully rigorous proof is not required.]

[**Solution:** This is Exercise 2 **(c)** on homework set #3 from my Spring 2017 course; see the course page for solutions.]

5.4.3. Spanning forests

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. Of course, 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 using as few edges as possible. This is known as a "spanning forest":

Definition 5.4.5. 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 a connected multigraph, a spanning forest of G means the same as a spanning tree of G.

5.4.4. Existence and construction of a spanning tree

The following theorem is crucial, which is why we will outline four different proofs:

Theorem 5.4.6. Each connected multigraph G has at least one spanning tree.

First proof. Let G be a connected multigraph. We want to construct a spanning tree of G. We try to achieve this by removing edges from G one by one, until G becomes a tree. When doing so, we must be careful not to disconnect the graph (i.e., not to destroy its connectedness). According to Theorem 3.3.18, this can be achieved by making sure that we never remove a bridge (i.e., an edge that appears in no cycle). Thus, we keep removing non-bridges (i.e., edges that are not bridges) as long as we can (i.e., until we end up with a graph in which every edge is a bridge).

So here is the algorithm: We start with G, and we successively remove non-bridges one by one until we no longer have any non-bridges left[32]. This procedure cannot go on forever, since G has only finitely many edges. Thus, after finitely many steps, we will end up with a graph that has no non-bridges any more. This resulting graph therefore has no cycles (since any cycle would have at least one edge, and this edge would be a non-bridge), but is still connected (since G was connected, and we never lost connectedness as we removed only non-bridges). Thus, this resulting graph is a tree. Since it is also a spanning subgraph of G (by construction), it is therefore a spanning tree of G. This proves Theorem 5.4.6. □

Second proof (sketched). In the above first proof, we constructed a spanning tree of G by starting with G and successively removing edges until we got a tree. Now let us take the opposite strategy: Start with an empty graph on the same vertex set as G, and successively add edges (from G) until we get a connected graph.

Here are some details: We start with a graph L that has the same vertex set as G, but has no edges. Now, we inspect all edges e of G one by one (in some order). For each such edge e, we add it to L, but only if it does not create a cycle in L; otherwise, we discard this edge. Notice that adding an edge e

[32]**Warning:** We cannot remove several non-bridges at once! We have to remove them one by one. Indeed, if e and f are two non-bridges of G, then there is no guarantee that f remains a non-bridge in $G \setminus e$. So we cannot remove both e and f simultaneously; we have to remove one of them and check whether the other is still a non-bridge.

with endpoints u and v to L creates a cycle if and only if u and v lie in the same component of L (before we add e). Thus, we only add an edge to L if its endpoints lie in different components of L; otherwise, we discard it. This way, at the end of the procedure, our graph L will still have no cycles (since we never create any cycles). In other words, it will be a forest.

Let me denote this forest by H. (Thus, H is the L at the end of the procedure.) I claim that this forest H is a spanning tree of G. Why? Since we know that H is a forest, we only need to show that H is connected. Assume the contrary. Thus, there is at least one edge e of G whose endpoints lie in different components of H (why?). This edge e is therefore not an edge of H. Therefore, at some point during our construction of H, we must have discarded this edge e (instead of adding it to L). As we know, this means that the endpoints of e used to lie in the same component of L at the point at which we discarded e. But this entails that these two endpoints lie in the same component of L at the end of the procedure as well (because the graph L never loses any edges during the procedure, so that any two vertices that used to lie in the same component of L at some point will still lie in the same component of L ever after). In other words, the endpoints of e lie in the same component of H. This contradicts our assumption that the endpoints of e lie in different components of H. This contradiction completes our proof that H is connected. Hence, H is a spanning tree of G, and we have proved Theorem 5.4.6 again. $\square$

Third proof. This proof takes yet another approach to constructing a spanning tree of G: We choose an arbitrary vertex r of G, and then progressively "spread a rumor" from r. The rumor starts at vertex r. On day 0, only r has heard the rumor. Every day, every vertex that knows the rumor spreads it to all its neighbors (i.e., all vertices adjacent to it). Since G is connected, the rumor will eventually spread to every vertex of G. Now, each vertex v (other than r) remembers which other vertex v' it has first heard the rumor from (if it heard it from several vertices at the same time, it just picks one of them), and picks some edge e_v that has endpoints v and v' (such an edge must exist, since v must have heard the rumor from a neighbor). The edges e_v for all $v \in V \setminus \{r\}$ (where V is the vertex set of G) then form a spanning tree of G (that is, the graph with vertex set V and edge set $\{e_v \mid v \in V \setminus \{r\}\}$ is a spanning tree). Why?

Intuitively, this is quite convincing: This graph cannot have cycles (because that would require a time loop) and must be connected (because for any vertex v, we can trace back the path of the rumor from r to v by following the edges e_v backwards). To obtain a rigorous proof, we formalize this construction mathematically:

Write G as $G = (V, E, \varphi)$. Choose any vertex r of G.

We shall recursively construct a sequence of subgraphs

$$(V_0, E_0, \varphi_0), \quad (V_1, E_1, \varphi_1), \quad (V_2, E_2, \varphi_2), \quad \ldots$$

of G. The idea behind these subgraphs is that for each $i \in \mathbb{N}$, the set V_i will consist of all vertices v that have heard the rumor by day i, and the set E_i will consist of the corresponding edges e_v. The map φ_i will be the restriction of φ to E_i, of course.

Here is the exact construction of this sequence of subgraphs:

- *Recursion base:* Set $V_0 := \{r\}$ and $E_0 := \varnothing$. Let φ_0 be the restriction of φ to the (empty) set E_0.

- *Recursion step:* Let $i \in \mathbb{N}$. Assume that the subgraph (V_i, E_i, φ_i) of G has already been defined. Now, we set

$$V_{i+1} := V_i \cup \{v \in V \mid v \text{ is adjacent to some vertex in } V_i\}.$$

For each $v \in V_{i+1} \setminus V_i$, we choose **one** edge e_v that joins[33] v to a vertex in V_i (such an edge exists, since $v \in V_{i+1}$; if there are several, we just choose a random one). Set

$$E_{i+1} := E_i \cup \{e_v \mid v \in V_{i+1} \setminus V_i\}.$$

Finally, we let φ_{i+1} be the restriction of the map φ to the set E_{i+1}. This is a map from E_{i+1} to $\mathcal{P}_{1,2}(V_{i+1})$ (because any edge e_v with $v \in V_{i+1} \setminus V_i$ has one endpoint v in $V_{i+1} \setminus V_i \subseteq V_{i+1}$ and the other endpoint in $V_i \subseteq V_{i+1}$). Thus, $(V_{i+1}, E_{i+1}, \varphi_{i+1})$ is a well-defined subgraph of G.

This construction yields that (V_i, E_i, φ_i) is a subgraph of $(V_{i+1}, E_{i+1}, \varphi_{i+1})$ for each $i \in \mathbb{N}$. Hence, $V_0 \subseteq V_1 \subseteq V_2 \subseteq \cdots$, so that $|V_0| \leq |V_1| \leq |V_2| \leq \cdots$. Since a sequence of integers bounded from above cannot keep increasing forever (and the sizes $|V_i|$ are bounded from above by $|V|$, since each V_i is a subset of V), we thus see that there exists some $i \in \mathbb{N}$ such that $|V_i| = |V_{i+1}|$. Consider this i. From $|V_i| = |V_{i+1}|$, we obtain $V_i = V_{i+1}$ (since $V_i \subseteq V_{i+1}$).

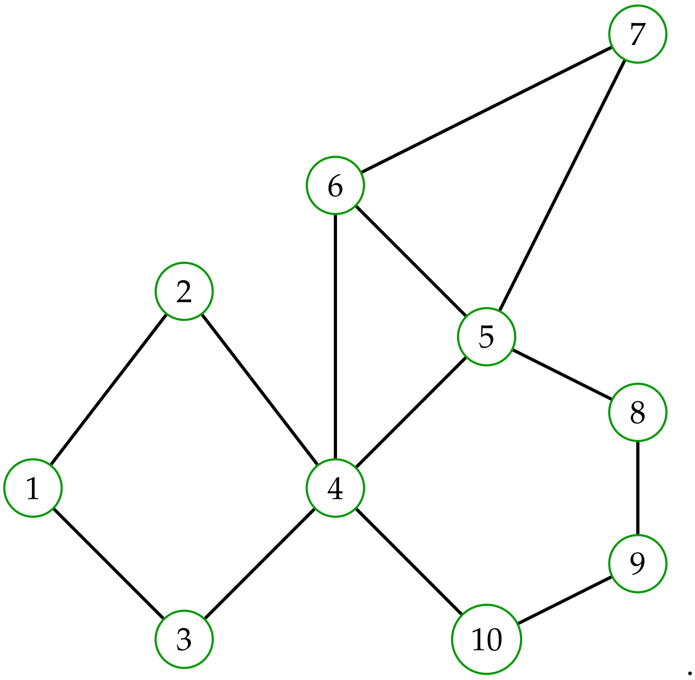
In our colloquial model above, $V_i = V_{i+1}$ means that no new vertices learn the rumor on day $i+1$; it is reasonable to expect that at this point, every vertex has heard the rumor. In other words, we claim that $V_i = V$. A rigorous proof of this can be easily given using the fact that G is connected[34].

Now, we claim that the subgraph (V_i, E_i, φ_i) is a spanning tree of G. To see this, we must show that this subgraph is a forest and is connected (since $V_i = V$ already shows that it is a spanning subgraph). Before we do this, let us give an example:

[33]We say that an edge **joins** a vertex p to a vertex q if the endpoints of this edge are p and q.

[34]Here is the *proof* in detail: We must show that $V_i = V$. Assume the contrary. Thus, there exists a vertex $u \in V \setminus V_i$. Consider this u. The path from r to u starts at a vertex in V_i (since $r \in V_0 \subseteq V_i$) and ends at a vertex in $V \setminus V_i$ (since $u \in V \setminus V_i$). Thus, it must cross over from V_i into $V \setminus V_i$ at some point. Therefore, there exists an edge with one endpoint in V_i and the other endpoint in $V \setminus V_i$. Let v and w be these two endpoints, so that $v \in V_i$ and $w \in V \setminus V_i$. Then, w is adjacent to some vertex in V_i (namely, to v), and therefore belongs to V_{i+1} (by the definition of V_{i+1}). Hence, $w \in V_{i+1} = V_i$. But this contradicts $w \notin V \setminus V_i$. This contradiction shows that our assumption was wrong, qed.

Example 5.4.7. Let G be the following multigraph:

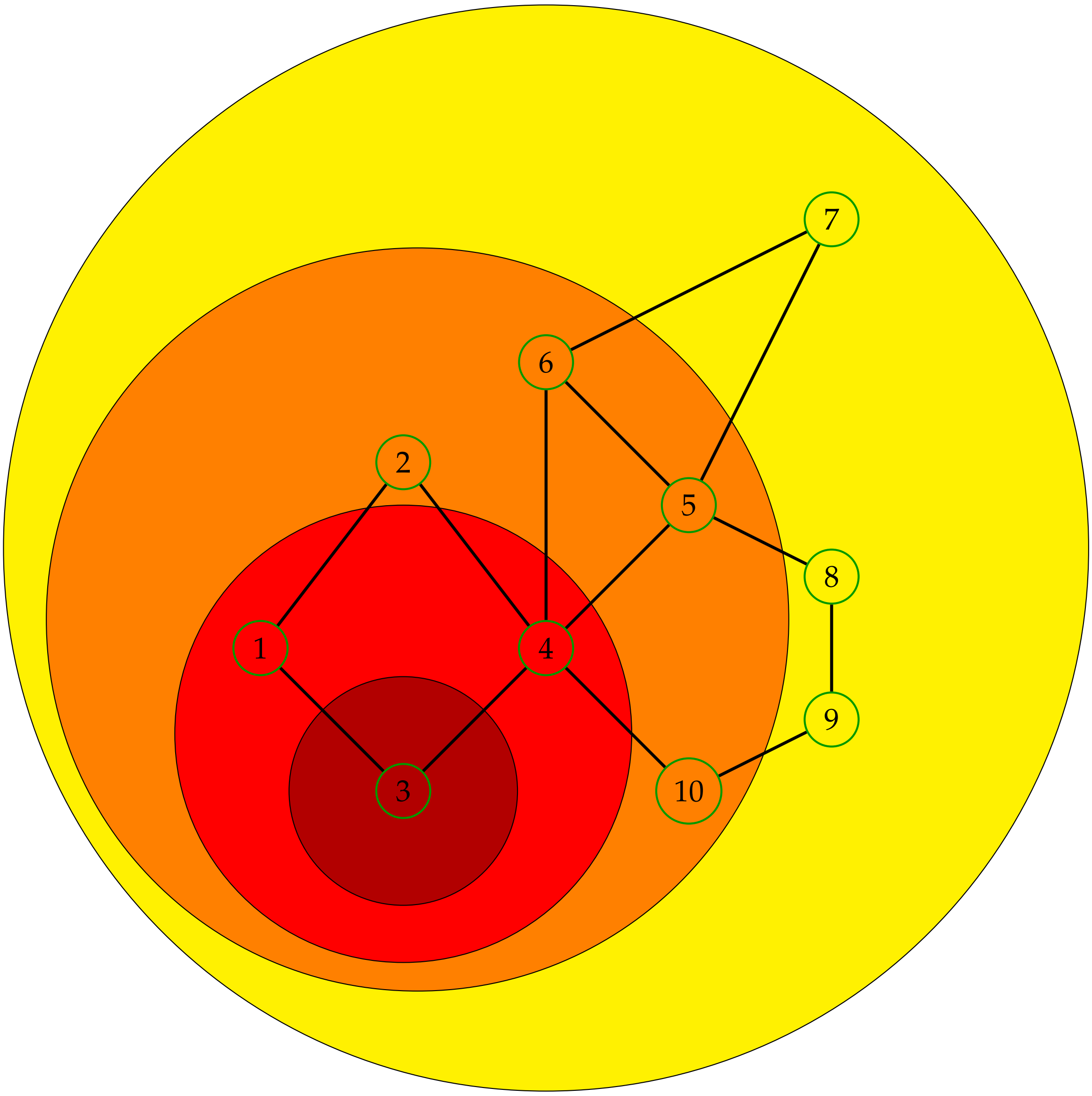


Set $r = 3$. Then, the above construction yields

$$\begin{aligned}
V_0 &= \{3\}, \\
V_1 &= \{3, 1, 4\}, \\
V_2 &= \{3, 1, 4, 2, 5, 6, 10\}, \\
V_3 &= \{3, 1, 4, 2, 5, 6, 10, 8, 9, 7\} = V,
\end{aligned}$$

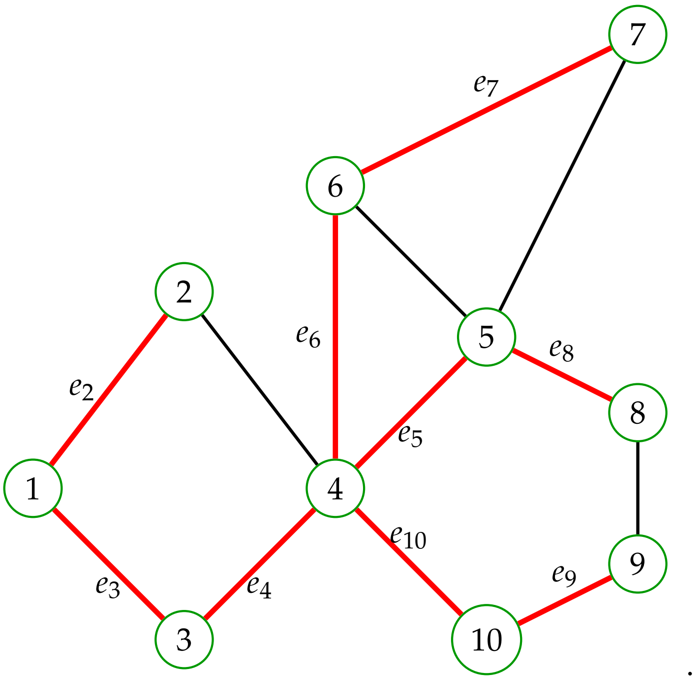
so that $V_k = V$ for all $k \geq 3$. Thus, we can take $i = 3$. Here is an image of the

V_k as progressively growing circles:



.

(The dark-red inner circle is V_0; the red circle is V_1; the orange circle is V_2; the yellow circle is $V_3 = V_4 = V_5 = \cdots = V$.) Finally, the edges e_v can be chosen to be the following (we are painting them red for clarity):



(Here, we have made two choices: We chose e_2 to be the edge joining 2 with 1 rather than the edge joining 2 with 4, and we chose e_7 to be the edge joining 7 with 6 rather than 7 with 5. The other options would have been equally fine.)

We now return to the general proof. Let us first show the following:

> *Claim 1:* Let $j \in \mathbb{N}$. Each vertex of the graph (V_j, E_j, φ_j) is path-connected to r in this graph.

[*Proof of Claim 1:* We induct on j:

Base case: For $j = 0$, Claim 1 is obvious, since $V_0 = \{r\}$ (so the only vertex of the graph in question is r itself).

Induction step: Fix some positive integer k. Assume (as the induction hypothesis) that Claim 1 holds for $j = k - 1$. That is, each vertex of the graph $(V_{k-1}, E_{k-1}, \varphi_{k-1})$ is path-connected to r in this graph.

Now, let v be a vertex of the graph (V_k, E_k, φ_k). We must show that v is path-connected to r in this graph. If $v \in V_{k-1}$, then this follows from the induction hypothesis (since $(V_{k-1}, E_{k-1}, \varphi_{k-1})$ is a subgraph of (V_k, E_k, φ_k)). Thus, we WLOG assume that $v \notin V_{k-1}$ from now on. Hence, $v \in V_k \setminus V_{k-1}$. According to the recursive definition of E_k, this entails that there is an edge $e_v \in E_k$ that joins v to some vertex $u \in V_{k-1}$. Consider this latter vertex u. Then, v is path-connected to u in the graph (V_k, E_k, φ_k) (since the edge e_v provides a length-1 path from v to u). However, u is path-connected to r in the graph

$(V_{k-1}, E_{k-1}, \varphi_{k-1})$ (by the induction hypothesis, since $u \in V_{k-1}$), hence also in the graph (V_k, E_k, φ_k) (since $(V_{k-1}, E_{k-1}, \varphi_{k-1})$ is a subgraph of (V_k, E_k, φ_k)). Since the relation "path-connected" is transitive, we conclude from the previous two sentences that v is path-connected to r in the graph (V_k, E_k, φ_k).

So we have shown that each vertex v of the graph (V_k, E_k, φ_k) is path-connected to r in the graph (V_k, E_k, φ_k). In other words, Claim 1 holds for $j = k$. This completes the induction step, and Claim 1 is proved.]

Claim 1 (applied to $j = i$) shows that each vertex of the graph (V_i, E_i, φ_i) is path-connected to r in this graph. Since the relation "path-connected" is an equivalence relation, this entails that any two vertices of this graph are path-connected. Thus, the graph (V_i, E_i, φ_i) is connected (since it has at least one vertex). It remains to prove that this graph (V_i, E_i, φ_i) is a forest.

Again, we do this using an auxiliary claim:

> *Claim 2:* Let $j \in \mathbb{N}$. Then, the graph $\left(V_j, E_j, \varphi_j\right)$ has no cycles.

[*Proof of Claim 2:* We induct on j:

Base case: The graph (V_0, E_0, φ_0) has no edges (because $E_0 = \varnothing$) and thus no cycles. Thus, Claim 2 holds for $j = 0$.

Induction step: Fix some positive integer k. Assume (as the induction hypothesis) that Claim 2 holds for $j = k - 1$. That is, the graph $(V_{k-1}, E_{k-1}, \varphi_{k-1})$ has no cycles.

Now, let C be a cycle of the graph (V_k, E_k, φ_k). Then, C must use at least one edge from $E_k \setminus E_{k-1}$ (since otherwise, C would be a cycle of the graph $(V_{k-1}, E_{k-1}, \varphi_{k-1})$, but this is impossible, since $(V_{k-1}, E_{k-1}, \varphi_{k-1})$ has no cycles). However, each edge from $E_k \setminus E_{k-1}$ has the form e_v for some $v \in V_k \setminus V_{k-1}$ (because of how E_k was defined). Thus, C must have an edge of this form. Consider the corresponding vertex $v \in V_k \setminus V_{k-1}$. The cycle C contains the edge e_v and therefore also contains its endpoint v. However, (again by the definition of E_k) the edge e_v is the **only** edge in E_k that contains the vertex v. Thus, the vertex v cannot be contained in any cycle of (V_k, E_k, φ_k) (because a cycle would necessarily include **two** distinct edges that contain v). This contradicts the fact that the cycle C contains v.

Forget that we fixed C. We thus have obtained a contradiction for each cycle C of the graph (V_k, E_k, φ_k). Hence, the graph (V_k, E_k, φ_k) has no cycles. In other words, Claim 2 holds for $j = k$. This completes the induction step, and Claim 2 is proved.]

Applying Claim 2 to $j = i$, we see that the graph (V_i, E_i, φ_i) has no cycles. In other words, this graph is a forest. Since it is connected, it is therefore a tree. Since it is a spanning subgraph of G, we thus conclude that it is a spanning tree of G. Hence, we have constructed a spanning tree of G.

We note an important property of this construction:

Claim 3: For each $k \in \mathbb{N}$, we have

$$V_k = \{v \in V \mid d(r, v) \leq k\},$$

where $d(r, v)$ means the length of a shortest path from r to v.

This is easily proved by induction on k. Thus, the spanning tree (V_i, E_i, φ_i) we have constructed has the following property: For each $v \in V$, the path from r to v in this spanning tree is a shortest path from r to v in G. For this reason, this spanning tree is called a **breadth-first search ("BFS") tree**. Note that the choice of root r is important here: It is usually not true that the path from an arbitrary vertex u to an arbitrary vertex v along our spanning tree is a shortest path in G. No spanning tree of G has this property, unless G itself is "more or less a tree" (more precisely, unless G^{simp} is a tree)! □

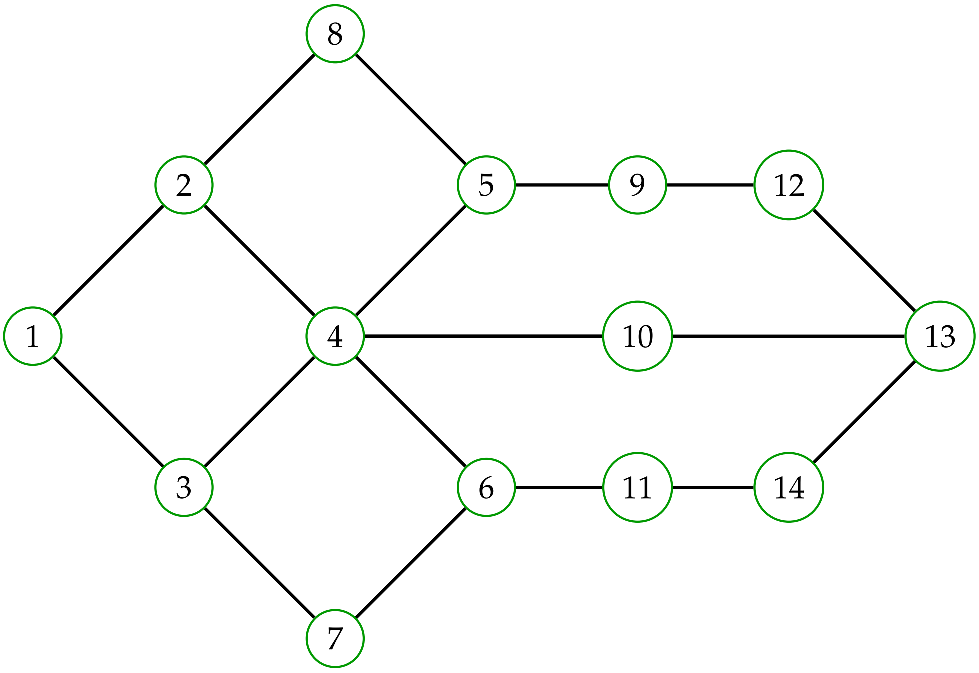
Fourth proof of Theorem 5.4.6 (sketched). We imagine a snake that slithers along the edges of G, trying to eventually bite each vertex. It starts at some vertex r, which it immediately bites. Any time the snake enters a vertex v, it makes the following step:

- If some neighbor of v has not been bitten yet, then the snake picks such a neighbor w as well as some edge f that joins w with v; the snake then moves to w along the edge f, bites the vertex w and marks the edge f.
- If not, then the snake marks the vertex v as fully digested and backtracks (along the marked edges) to the last vertex it has visited but not fully digested yet.

Once backtracking is no longer possible (because there are no more vertices left that are not fully digested), the procedure is finished. I claim that the marked edges at that moment are the edges of a spanning tree of G.

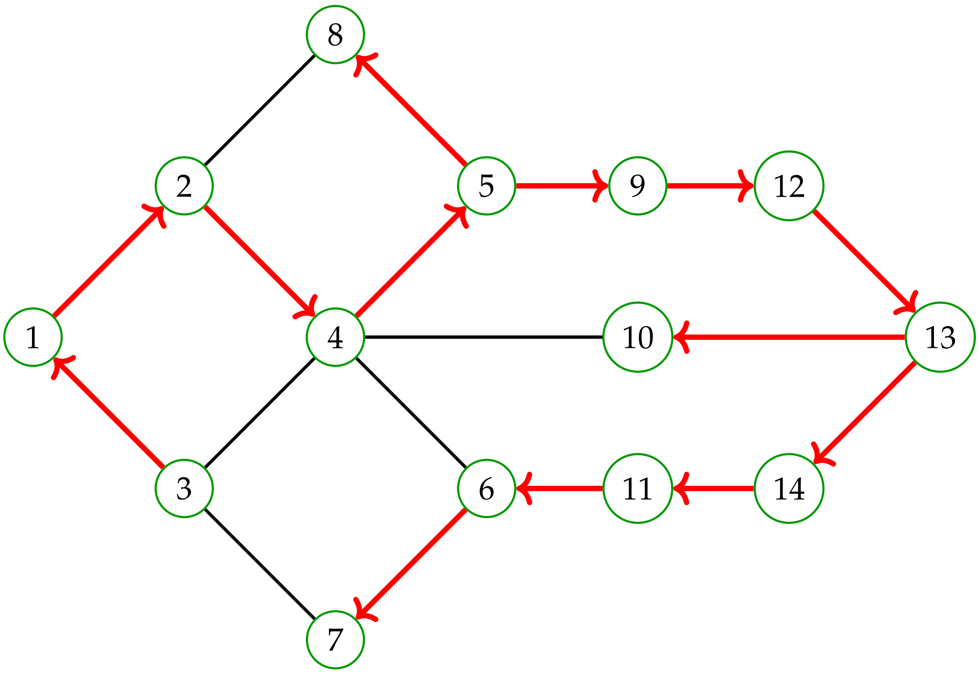
I won't prove this claim in detail, but I will give some hints. First, however, an example:

Example 5.4.8. Let G be the following connected multigraph:



.

Let our snake start its journey at $r = 3$. It bites this vertex. Then, let's say that it picks the vertex 1 as its next victim (it could just as well go to 4 or 7; the snake has many choices, but we follow one possible trip). Thus, it next arrives at vertex 1, bites it and marks the edge that brought it to this vertex. As its next destination, it necessarily picks the vertex 2 (since vertex 3 has already been bitten). It moves to vertex 2, bites it and marks the edge. Next, let's say that it picks the vertex 4 (the other option would be 8). It thus moves to 4, bites it and marks the edge. Proceeding likewise, it then moves to 5 (the other options are 6 and 10; the vertices 2 and 3 do not qualify since they are already bitten), bites 5 and marks an edge. From there, let's say it moves to 8, bites 8 and marks an edge. Now, there is no longer an unbitten neighbor of 8 to move to. Thus, the snake marks the vertex 8 as fully digested and backtracks to the last vertex not fully digested – which, at this point, is 5. From this vertex 5, it moves on to 9 (this is the only option, since 4 and 8 have already been bitten). And so on. Here is one possible outcome of this journey (there are a few more decisions that the snake can make here, so you may get a different one):



Here, the marked edges are drawn in bold red ink, and endowed with an arrow that represents the direction in which they were first used (e.g., the edge joining 2 with 4 has an arrow towards 4 because it was first used to get from 2 to 4).

Now, as promised, let me outline a proof of the above claim (that the marked edges form a spanning tree of G). To wit, argue the following four observations (ideally in this order):

1. After each step, the marked edges are precisely the edges along which the snake has moved so far.

2. After each step, the network of bitten vertices and marked edges is a tree.

3. After enough steps, each bitten vertex is fully digested.

4. At that point, the network of bitten vertices and marked edges is a spanning tree (since each neighbor of a fully digested vertex is bitten, thus fully digested by observation 3).

Details are left to the reader.

The result is that Theorem 5.4.6 is proved once again. However, more comes out of the above construction if you know where to look. The spanning tree T of G whose edges are the edges marked by the snake is called a **depth-first search ("DFS") tree**. It has the following extra property: If u and v are two adjacent vertices of G, then either u lies on the path from r to v in T, or v lies on the path from r to u in T. (This called a "lineal spanning tree". See [BenWil06, §6.1] for details.) □

5.4.5. 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 in particular provides "canonical" paths between any two vertices. This is useful, e.g., for networking applications where having a choice between different paths would be problematic (see, e.g., the Spanning Tree Protocol).

- A w-minimum spanning tree (see Exercise 5.8 = Homework set #5 exercise 6) solves a global version of the cheapest-path problem. It can also be used for detecting clusters.

- Depth-first search (the algorithm used in our fourth proof of Theorem 5.4.6) can also be used as a way to traverse all vertices of a given graph and return back to the starting point. In particular, this provides an algorithmic way to solve mazes (since a maze can be modeled as a graph, where the vertices correspond to "rooms" and the edges correspond to "doors"). This appears to have been the original motivation for Trémaux to invent depth-first search back in the 19th century.

Here is a more theoretical application of spanning trees:

Definition 5.4.9. A vertex v of a connected multigraph G is said to be a **cut-vertex** if the graph $G \setminus v$ is disconnected. (Recall that $G \setminus v$ is the multigraph obtained from G by removing the vertex v and all edges that contain v.)

Proposition 5.4.10. Let G be a connected multigraph with ≥ 2 vertices. Then, there are at least 2 vertices of G that are **not** cut-vertices.

Proof. Pick a spanning tree T of G (we know from Theorem 5.4.6 that such a spanning tree exists). Then, T has at least 2 leaves (by Theorem 5.3.2 **(a)**). But each leaf of T is a non-cut-vertex of G (why?). □

Remark 5.4.11. It is not true that conversely, any non-leaf of T is a cut-vertex of G. So we cannot get any lower bound on the number of cut-vertices. And this is not surprising: Lots of graphs (e.g., the complete graph K_n for $n \geq 2$) have no cut-vertices at all. These graphs are said to be **2-connected**, and their properties have been amply studied (see, e.g., [West01, §4.2] for an introduction).

5.4.6. Exercises

Exercise 5.6. Let G be a connected multigraph. Let T_1 and T_2 be two spanning trees of G.

Prove the following:[35]

(a) For any $e \in \operatorname{E}(T_1) \setminus \operatorname{E}(T_2)$, there exists an $f \in \operatorname{E}(T_2) \setminus \operatorname{E}(T_1)$ with the property that replacing e by f in T_1 (that is, removing the edge e from T_1 and adding the edge f) results in a spanning tree of G.

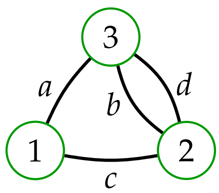
(b) For any $f \in \operatorname{E}(T_2) \setminus \operatorname{E}(T_1)$, there exists an $e \in \operatorname{E}(T_1) \setminus \operatorname{E}(T_2)$, with the property that replacing e by f in T_1 (that is, removing the edge e from T_1 and adding the edge f) results in a spanning tree of G.

[**Hint:** The two parts look very similar, but (to my knowledge) their proofs are not.]

Exercise 5.7. Let G be a connected multigraph. Let $\mathcal{S}$ be the simple graph whose vertices are the spanning trees of G, and whose edges are defined as follows: Two spanning trees T_1 and T_2 of G are adjacent (as vertices of $\mathcal{S}$) if and only if T_2 can be obtained from T_1 by removing an edge and adding another (i.e., if and only if there exist an edge e_1 of T_1 and an edge e_2 of T_2 such that $e_2 \neq e_1$ and $T_2 \setminus e_2 = T_1 \setminus e_1$).

Prove that the simple graph $\mathcal{S}$ is itself connected. (In simpler language: Prove that any spanning tree of G can be transformed into any other spanning tree of G by a sequence of legal "remove an edge and add another" operations, where such an operation is called **legal** if its result is a spanning tree of G.)

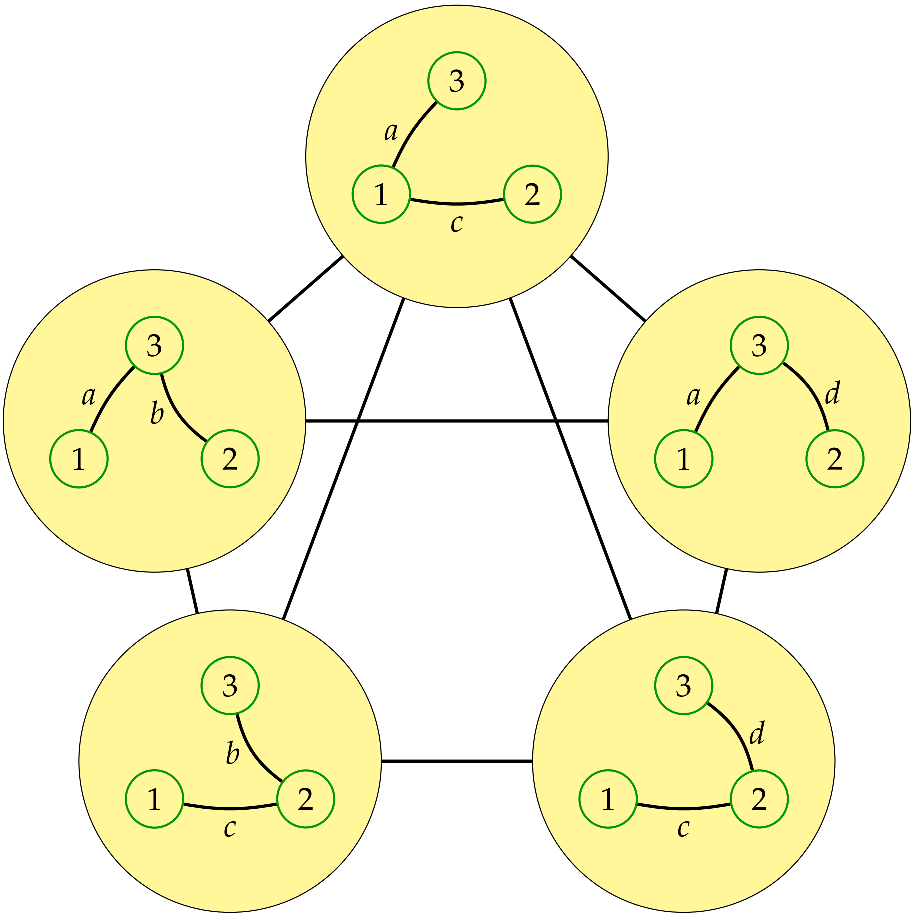
[**Example:** If G is the multigraph



,

[35] Recall that $\operatorname{E}(H)$ denotes the edge set of any graph H.

then the graph $\mathcal{S}$ looks as follows:



]

Exercise 5.8. Let $G = (V, E, \varphi)$ be a connected multigraph. Let $w : E \to \mathbb{R}$ be a map that assigns a real number $w(e)$ to each edge e. We shall call this real number $w(e)$ the **weight** of the edge e.

If $H = (W, F, \varphi\mid_F)$ is a subgraph of G, then the **weight** $w(H)$ of H is defined to be $\sum_{f \in F} w(f)$ (that is, the sum of the weights of all edges of H).

A w**-minimum spanning tree** of G means a spanning tree of G that has the smallest weight among all spanning trees of G.

In our first proof of Theorem 5.4.6, we have seen a way to construct a spanning tree of G by successively removing non-bridges until only bridges remain. (A **non-bridge** means an edge that is not a bridge.)

Now, let us perform this algorithm, but taking care to choose a non-bridge of largest weight (among all non-bridges) at each step. Prove that the result will be a w-minimum spanning tree.

Exercise 5.9. Let G be a connected multigraph with an even number of vertices. Prove that there exists a spanning subgraph H of G such that each vertex of H has odd degree (in H).

[Hint: One way to solve this begins by reducing the problem to the case when G is a tree.]

5.4.7. Existence and construction of a spanning forest

So we have learnt that connected graphs have spanning trees. What do disconnected graphs have?

Corollary 5.4.12. Each multigraph has a spanning forest.

Proof. Apply Theorem 5.4.6 to each component of the multigraph. Then, combine the resulting spanning trees into a spanning forest. □

5.5. Centers of graphs and trees

5.5.1. Distances

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

Definition 5.5.1. Let G be a multigraph.

For any two vertices u and v of G, we define the **distance** between u and v to be the smallest length of a path from u to v. If no such path exists, then this distance is defined to be ∞.

The distance between u and v is denoted by $d(u,v)$ or by $d_G(u,v)$ when the graph G is not clear from the context.

Example 5.5.2. If G is the multigraph from Example 5.4.8, then

$$d_G(1,9) = 4, \qquad d_G(4,13) = 2, \qquad d_G(4,4) = 0.$$

Remark 5.5.3. Distances in a multigraph satisfy the rules that you would expect a distance function to satisfy:

(a) We have $d(u,u) = 0$ for any vertex u.

(b) We have $d(u,v) = d(v,u)$ for any vertices u and v.

(c) We have $d(u,v) + d(v,w) \geq d(u,w)$ for any vertices u, v and w. (Here, we understand that $\infty \geq m$ and $\infty + m = \infty$ for any $m \in \mathbb{N}$.)

Also:

(d) The distances $d(u,v)$ do not change if we replace "path" by "walk" in the definition of the distance.

(e) If V is the vertex set of our multigraph, then $d(u,v) \leq |V| - 1$ for any vertices u and v.

Proof. Part **(d)** follows from Corollary 3.3.10. The proofs of **(a)**, **(b)** and **(c)** are then straightforward (the proof of **(c)** relies on part **(d)**, because splicing two paths generally only yields a walk, not a path). Finally, in order to prove part **(e)**, observe that any path of our multigraph has length $\leq |V| - 1$ (since its vertices are distinct). $\square$

We note that the definition of a distance becomes simpler if our multigraph is a tree: Namely, if T is a tree, then the distance $d(u,v)$ between two vertices u and v is the length of the **only** path from u to v in T. Thus, in a tree, we do not have to worry whether a given path is the shortest.

We also notice that if G is a multigraph, and if u and v are two vertices of G, then the distance $d_G(u,v)$ in G equals the distance $d_{G^{\text{simp}}}(u,v)$ in the simple graph G^{simp}. (The reason for this is that any path of G can be converted into a path of G^{simp} having the same length, and vice versa. Of course, this is not a one-to-one correspondence, but it suffices for our purposes.) Thus, when studying distances on a multigraph, we can WLOG restrict ourselves to simple graphs.

The following few exercises give some curious properties of distances in various kinds of graphs.

Exercise 5.10. Let a, b and c be three vertices of a connected multigraph $G = (V, E, \varphi)$. Prove that $d(b,c) + d(c,a) + d(a,b) \leq 2|V| - 2$.

[**Solution:** This is Exercise 7 on midterm #1 from my Spring 2017 course, except that the simple graph has been replaced by a multigraph (but this makes no serious difference); see the course page for solutions.]

Exercise 5.11. Let a, b and c be three vertices of a strongly connected multidigraph $D = (V, A, \psi)$ such that $|V| \geq 4$. For any two vertices u and v of D, we define the distance $d(u,v)$ to be the smallest length of a path from u to v. (This definition is the obvious analogue of Definition 5.5.1 for digraphs.)

(a) Prove that $d(b,c) + d(c,a) + d(a,b) \leq 3|V| - 4$.

(b) For each $n \geq 5$, construct an example in which $|V| = n$ and $d(b,c) + d(c,a) + d(a,b) = 3|V| - 4$. (No proof is required for the example.)

[**Solution:** This is Exercise 5 on homework set #3 from my Spring 2017 course, except that the simple digraph has been replaced by a multidigraph (but this makes no serious difference); see the course page for solutions.]

Exercise 5.12. Let G be a tree. Let x, y, z and w be four vertices of G.

Show that the two largest ones among the three numbers

$$d(x,y) + d(z,w), \qquad d(x,z) + d(y,w) \qquad \text{and} \qquad d(x,w) + d(y,z)$$

are equal.

[**Solution:** This is Exercise 6 on midterm #2 from my Spring 2017 course; see the course page for solutions.]

Exercise 5.13. Let G be a connected multigraph. Let x, y, z and w be four vertices of G.

Assume that the two largest ones among the three numbers

$$d(x,y) + d(z,w), \qquad d(x,z) + d(y,w) \qquad \text{and} \qquad d(x,w) + d(y,z)$$

are **not** equal.

Prove that G has a cycle of length $\leq d(x,z) + d(y,w) + d(x,w) + d(y,z)$.

[**Hint:** This is a strengthening of Exercise 5.12. Try deriving it by applying the latter exercise to a strategically chosen subgraph of G.]

[**Solution:** This is Exercise 1 on midterm #3 from my Spring 2017 course; see the course page for solutions.]

5.5.2. Eccentricity and centers

We can now define "eccentricities":

Definition 5.5.4. Let v be a vertex of a multigraph $G = (V, E, \varphi)$. The **eccentricity** of v (with respect to G) is defined to be the number

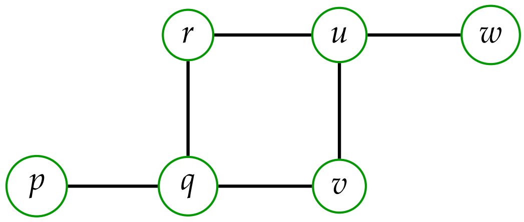
$$\max \{d(v,u) \mid u \in V\} \in \mathbb{N} \cup \{\infty\}.$$

This eccentricity is denoted by $\operatorname{ecc} v$ or $\operatorname{ecc}_G v$.

Definition 5.5.5. Let $G = (V, E, \varphi)$ be a multigraph. Then, a **center** of G means a vertex of G whose eccentricity is minimum (among all vertices).

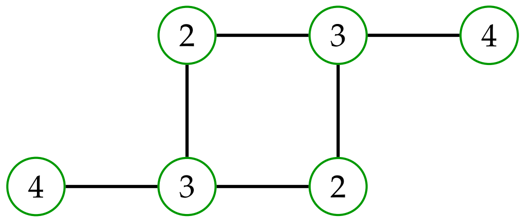
(Some authors have a slightly different definition of a "center": They define the **center** of G to be the **set** of all vertices of G whose eccentricity is minimum. That is, what they call "center" is the set of what we call "centers".)

Example 5.5.6. Let G be the following multigraph:



.

Then, the eccentricities of its vertices are as follows (we are just labeling each vertex with its eccentricity):



.

Thus, the centers of G are the vertices r and v.

Example 5.5.7. Let G be a complete graph K_n (with n vertices). Then, each vertex of G has the same eccentricity (which is 1 if $n \geq 2$ and 0 if $n = 1$), and thus each vertex of G is a center of G.

Example 5.5.8. Let G be a graph with more than one component. Then, each vertex v of G has eccentricity ∞ (because there exists at least one vertex u that lies in a different component of G than v, and thus this vertex u satisfies $d(v, u) = \infty$). Hence, each vertex of G is a center of G.

5.5.3. The centers of a tree

As we see from Example 5.5.8, eccentricity and centers are not very useful notions when the graph is disconnected. Even for a connected graph, Example 5.5.6 shows that the centers do not necessarily form a connected subgraph. However, in a tree, they behave a lot better:

Theorem 5.5.9. Let T be a tree. Then:

(a) The tree T has either 1 or 2 centers.

(b) If T has 2 centers, then these 2 centers are adjacent.

(c) Moreover, these centers can be found by the following algorithm:

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, we remove all leaves from it (simultaneously). The result is again a tree. If that tree still has more than 2 vertices, we remove all leaves from it (simultaneously), and continue doing so until we are left with a tree that has only 1 or 2 vertices. These vertices are the centers of T.

To prove Theorem 5.5.9, we first study how a tree is affected when all its leaves are removed:

Lemma 5.5.10. Let $T = (V, E, \varphi)$ be a tree with more than 2 vertices.

Let L be the set of all leaves of T.

Let $T \setminus L$ be the induced submultigraph of T on the set $V \setminus L$. (Thus, $T \setminus L$ is obtained from T by removing all the vertices in L and all adjacent that contain a vertex in L.)

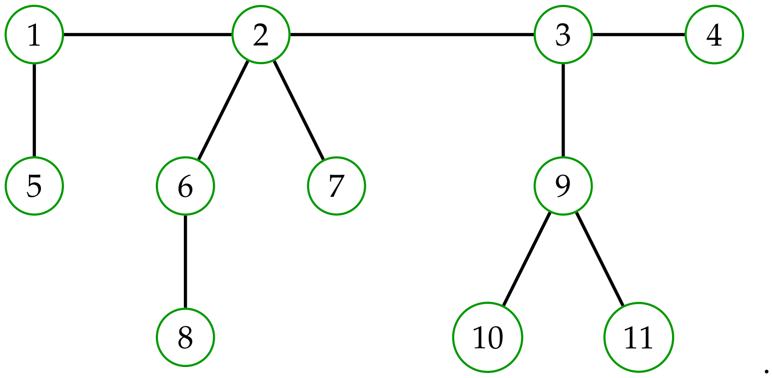
Then:

- **(a)** The multigraph $T \setminus L$ is a tree.
- **(b)** For any $u \in V \setminus L$ and $v \in V \setminus L$, we have

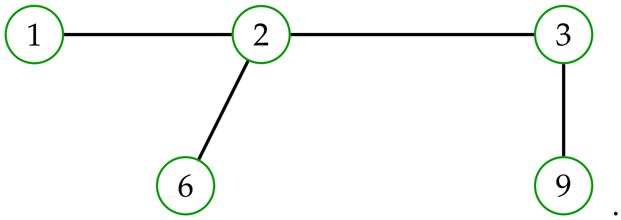
$$\{\text{paths of } T \text{ from } u \text{ to } v\} = \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\}$$

  (that is, the paths of T from u to v are precisely the paths of $T \setminus L$ from u to v).
- **(c)** For any $u \in V \setminus L$ and $v \in V \setminus L$, we have $d_T(u, v) = d_{T \setminus L}(u, v)$.
- **(d)** Each vertex $v \in V \setminus L$ satisfies $\operatorname{ecc}_T v = \operatorname{ecc}_{T \setminus L} v + 1$.
- **(e)** Each leaf $v \in L$ satisfies $\operatorname{ecc}_T v = \operatorname{ecc}_T w + 1$, where w is the unique neighbor of v in T. (A **neighbor** of v means a vertex that is adjacent to v.)
- **(f)** The centers of T are precisely the centers of $T \setminus L$.

Example 5.5.11. Let T be the following tree:



Then, the set L from Lemma 5.5.10 is $\{4, 5, 7, 8, 10, 11\}$, and the tree $T \setminus L$ looks as follows:



Proof of Lemma 5.5.10. First, we notice that T is a forest (since T is a tree), and thus has no cycles. In particular, T therefore has no loops and no parallel edges. Also, for any two vertices u and v of T, there is a unique path from u to v in T.

Next, we introduce some terminology: If $\mathbf{p}$ is a path of some multigraph, then an **intermediate vertex** of $\mathbf{p}$ shall mean a vertex of $\mathbf{p}$ that is neither the starting point nor the ending point of $\mathbf{p}$. In other words, if $\mathbf{p} = (p_0, e_1, p_1, e_2, p_2, \ldots, e_k, p_k)$ is a path of some multigraph, then the intermediate vertices of $\mathbf{p}$ are $p_1, p_2, \ldots, p_{k-1}$. Clearly, any intermediate vertex of a path $\mathbf{p}$ must have degree ≥ 2 (since the path $\mathbf{p}$ enters it along some edge, and leaves it along another). Hence, if $\mathbf{p}$ is a path of T, then

$$\text{any intermediate vertex of } \mathbf{p} \text{ must belong to } V \setminus L \tag{12}$$

(because it must have degree ≥ 2, thus cannot be a leaf of T; but this means that it cannot belong to L; therefore, it must belong to $V \setminus L$).

(b) Let $u \in V \setminus L$ and $v \in V \setminus L$. Let $\mathbf{p}$ be a path of T from u to v. We shall show that $\mathbf{p}$ is a path of $T \setminus L$ as well.

Indeed, let us first check that all vertices of $\mathbf{p}$ belong to $V \setminus L$. This is clear for the vertices u and v (since $u \in V \setminus L$ and $v \in V \setminus L$); but it also holds for every intermediate vertex of $\mathbf{p}$ (by (12)). Thus, it does indeed hold for all vertices of $\mathbf{p}$.

We have thus shown that all vertices of $\mathbf{p}$ belong to $V \setminus L$. Hence, $\mathbf{p}$ is a path of $T \setminus L$ (since $T \setminus L$ is the induced submultigraph of T on the set $V \setminus L$).

Forget that we fixed $\mathbf{p}$. We have thus shown that every path $\mathbf{p}$ of T from u to v is also a path of $T \setminus L$. Hence,

$$\{\text{paths of } T \text{ from } u \text{ to } v\} \subseteq \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} .$$

Conversely, we have

$$\{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} \subseteq \{\text{paths of } T \text{ from } u \text{ to } v\} ,$$

since every path of $T \setminus L$ is a path from T (because $T \setminus L$ is a submultigraph of T). Combining these two facts, we obtain

$$\{\text{paths of } T \text{ from } u \text{ to } v\} = \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} .$$

This proves Lemma 5.5.10 **(b)**.

(c) This follows from Lemma 5.5.10 **(b)**, since the distance $d_G(u, v)$ of two vertices u and v in a graph G is defined to be the smallest length of a path from u to v.

(a) The graph T is a tree, thus a forest. Hence, its submultigraph $T \setminus L$ is a forest as well (since any cycle of $T \setminus L$ would be a cycle of T). It thus remains to show that $T \setminus L$ is connected.

First, it is easy to see that $T \setminus L$ has at least one vertex[36]. It remains to show that any two vertices of $T \setminus L$ are path-connected.

Let u and v be two vertices of $T \setminus L$. Then, $u \in V \setminus L$ and $v \in V \setminus L$. Hence, Lemma 5.5.10 **(b)** yields

$$\{\text{paths of } T \text{ from } u \text{ to } v\} = \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} .$$

Thus, $\{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} = \{\text{paths of } T \text{ from } u \text{ to } v\} \neq \varnothing$ (since there exists a path of T from u to v (because T is connected)). In other words, there exists a path of $T \setminus L$ from u to v. In other words, u and v are path-connected in $T \setminus L$.

We have now shown that any two vertices u and v of $T \setminus L$ are path-connected in $T \setminus L$. This entails that $T \setminus L$ is connected (since $T \setminus L$ has at least one vertex). This proves Lemma 5.5.10 **(a)**.

[36] *Proof.* We assumed that T has more than 2 vertices. In other words, there exist three distinct vertices u, v, w of T. Consider these u, v, w. If all three distances $d_T(u, v)$, $d_T(v, w)$ and $d_T(w, u)$ were equal to 1, then T would have a cycle (of the form $(u, *, v, *, w, *, u)$, where each asterisk stands for some edge); but this would contradict the fact that T has no cycles. Thus, not all of these three distances are equal to 1. Hence, at least one of them is $\neq 1$. WLOG assume that $d_T(u, v) \neq 1$ (otherwise, we permute u, v, w). Hence, the path from u to v has more than one edge (indeed, it must have at least one edge, since u and v are distinct). Therefore, this path has at least one intermediate vertex. This intermediate vertex then must belong to $V \setminus L$ (by (12)). Hence, it is a vertex of the subgraph $T \setminus L$. This shows that $T \setminus L$ has at least one vertex.

(d) If u and v are two vertices of $T \setminus L$, then the two distances $d_T(u, v)$ and $d_{T \setminus L}(u, v)$ are equal (by Lemma 5.5.10 **(c)**); thus, we shall denote both distances by $d(u, v)$ (since there is no confusion to be afraid of).

Let $v \in V \setminus L$. We must show that $\operatorname{ecc}_T v = \operatorname{ecc}_{T \setminus L} v + 1$.

Let u be a vertex of $T \setminus L$ such that $d(v, u)$ is maximum. Thus, $\operatorname{ecc}_{T \setminus L} v = d(v, u)$ (by the definition of $\operatorname{ecc}_{T \setminus L} v$). However, u is a vertex of $T \setminus L$, and thus does not belong to L. Hence, u is not a leaf of T (since L is the set of all leaves of T). Hence, u has degree ≥ 2 in T (since a vertex in a tree with more than 1 vertex cannot have degree 0).

Now, consider the path $\mathbf{p}$ from v to u in the tree T. This path $\mathbf{p}$ has length $d(v, u)$. Since u has degree ≥ 2, there exist at least two edges of T that contain u. Hence, in particular, there exists at least one edge f that contains u and is distinct from the last edge of $\mathbf{p}$ [37]. Consider this edge f. Let w be the endpoint of f other than u. Appending f and w to the end of the path $\mathbf{p}$, we obtain a walk from v to w. This walk is backtrack-free (since f is distinct from the last edge of $\mathbf{p}$) and thus must be a path (by Proposition 5.1.2, since T has no cycles). This path has length $d(v, u) + 1$ (since it was obtained by appending an edge to the path $\mathbf{p}$, which has length $d(v, u)$). Hence, $d(v, w) = d(v, u) + 1$. But the definition of eccentricity yields

$$\operatorname{ecc}_T v \geq d(v, w) = \underbrace{d(v, u)}_{= \operatorname{ecc}_{T \setminus L} v} + 1 = \operatorname{ecc}_{T \setminus L} v + 1. \tag{13}$$

On the other hand, let x be a vertex of T such that $d(v, x)$ is maximum. Thus, $\operatorname{ecc}_T v = d(v, x)$ (by the definition of $\operatorname{ecc}_T v$). The path from v to x has length ≥ 1 (since otherwise, we would have $x = v$ and therefore $d(v, x) = d(v, v) = 0$, which would easily contradict the maximality of $d(v, x)$). Thus, it has a second-to-last vertex. Let y be this second-to-last vertex. Then, the path from v to y is simply the path from v to x with its last edge removed. Consequently, $d(v, y) = d(v, x) - 1$. However, it is easy to see that $y \in V \setminus L$ [38]. In other words, y is a vertex of $T \setminus L$. Thus, the definition of eccentricity yields

$$\operatorname{ecc}_{T \setminus L} v \geq d(v, y) = \underbrace{d(v, x)}_{= \operatorname{ecc}_T v} - 1 = \operatorname{ecc}_T v - 1,$$

so that $\operatorname{ecc}_T v \leq \operatorname{ecc}_{T \setminus L} v + 1$. Combining this with (13), we obtain $\operatorname{ecc}_T v = \operatorname{ecc}_{T \setminus L} v + 1$. This proves Lemma 5.5.10 **(d)**.

[37] If the path $\mathbf{p}$ has no edges, then f can be any edge that contains u.

[38] *Proof.* Assume the contrary. Thus, $y \notin V \setminus L$. Hence, $y \neq v$ (since $y \notin V \setminus L$ but $v \in V \setminus L$).

However, y is the second-to-last vertex of the path from v to x. Therefore, y is either the starting point v of this path, or an intermediate vertex of this path. Since $y \neq v$, we thus conclude that y is an intermediate vertex of this path. Hence, by (12), we see that y must belong to $V \setminus L$. But this contradicts $y \notin V \setminus L$. This contradiction shows that our assumption was false, qed.

(e) If u and v are two vertices of $T \setminus L$, then the two distances $d_T(u,v)$ and $d_{T\setminus L}(u,v)$ are equal (by Lemma 5.5.10 **(c)**); thus, we shall denote both distances by $d(u,v)$ (since there is no confusion to be afraid of).

Let $v \in L$ be a leaf. Let w be the unique neighbor of v in T. We must prove that $\operatorname{ecc}_T v = \operatorname{ecc}_T w + 1$.

We first claim that

$$d(v,u) = d(w,u) + 1 \qquad \text{for each } u \in V \setminus \{v\}. \tag{14}$$

[*Proof of (14):* We have $\deg v = 1$ (since v is a leaf). In other words, there is a unique edge of T that contains v. Let e be this edge. The endpoints of e are v and w (since w is the unique neighbor of v). Thus, $v \neq w$ (since T has no loops) and $d(v,w) = 1$.

Now, let $u \in V \setminus \{v\}$. Then, the path from v to u in T must have length ≥ 1 (since $u \neq v$), and therefore must begin with the edge e (since e is the only edge that contains v). If we remove this edge e from this path, we thus obtain a path from w to u. As a consequence, the path from v to u is longer by exactly 1 edge than the path from w to u. In other words, we have $d(v,u) = d(w,u) + 1$. This proves (14).]

Now, the definition of eccentricity yields

$$\operatorname{ecc}_T v = \max\{d(v,u) \mid u \in V\}. \tag{15}$$

This maximum is clearly **not** attained for $u = v$ (since $d(v,v) = 0$ is smaller than $d(v,w) = 1$). Thus, this maximum does not change if we remove v from its indexing set V. Hence, (15) rewrites as

$$\begin{aligned}
\operatorname{ecc}_T v &= \max\left\{ \underbrace{d(v,u)}_{\substack{=d(w,u)+1 \\ \text{(by (14))}}} \mid u \in V \setminus \{v\} \right\} \\
&= \max\{d(w,u) + 1 \mid u \in V \setminus \{v\}\} \\
&= \max\{d(w,u) \mid u \in V \setminus \{v\}\} + 1.
\end{aligned} \tag{16}$$

On the other hand, the definition of eccentricity yields

$$\operatorname{ecc}_T w = \max\{d(w,u) \mid u \in V\}. \tag{17}$$

We shall now show that this maximum does not change if we remove v from its indexing set V. In other words, we shall show that

$$\max\{d(w,u) \mid u \in V\} = \max\{d(w,u) \mid u \in V \setminus \{v\}\}. \tag{18}$$

[*Proof of (18):* Assume that (18) is false. Then, the maximum $\max\{d(w,u) \mid u \in V\}$ is attained **only** at $u = v$. In other words, we have

$$d(w,v) > d(w,u) \qquad \text{for all } u \in V \setminus \{v\}. \tag{19}$$

However, the tree T has more than 2 vertices. Thus, it has a vertex u that is distinct from both v and w. Consider this u. Thus, $u \in V \setminus \{v\}$, so that (19) yields $d(w,v) > d(w,u)$. In view of $d(w,v) = d(v,w) = 1$, this rewrites as $1 > d(w,u)$, so that $d(w,u) < 1$. Therefore, $w = u$. But this contradicts the facts that w is distinct from u. This contradiction shows that our assumption was false, and thus (18) is proved.]

Now, (16) becomes

$$\begin{aligned}\operatorname{ecc}_T v &= \underbrace{\max\{d(w,u) \mid u \in V \setminus \{v\}\}}_{\substack{=\max\{d(w,u) \mid u \in V\} \\ \text{(by (18))}}} + 1 \\ &= \underbrace{\max\{d(w,u) \mid u \in V\}}_{\substack{=\operatorname{ecc}_T w \\ \text{(by (17))}}} + 1 = \operatorname{ecc}_T w + 1.\end{aligned}$$

This proves Lemma 5.5.10 **(e)**.

(f) Lemma 5.5.10 **(e)** shows that any vertex $v \in L$ has a higher eccentricity than its unique neighbor. Thus, a vertex v of T that minimizes $\operatorname{ecc}_T v$ cannot belong to L. In other words, a vertex v of T that minimizes $\operatorname{ecc}_T v$ must belong to $V \setminus L$.

However, the centers of T are defined to be the vertices of T that minimize $\operatorname{ecc}_T v$. As we just proved, these vertices must belong to $V \setminus L$. Thus, the centers of T can also be characterized as the vertices $v \in V \setminus L$ that minimize $\operatorname{ecc}_T v$. However, a vertex $v \in V \setminus L$ minimizes $\operatorname{ecc}_T v$ if and only if it minimizes $\operatorname{ecc}_{T\setminus L} v$ (because Lemma 5.5.10 **(d)** yields $\operatorname{ecc}_T v = \operatorname{ecc}_{T\setminus L} v + 1$ for any such vertex v). Thus, we conclude that the centers of T can be characterized as the vertices $v \in V \setminus L$ that minimize $\operatorname{ecc}_{T\setminus L} v$. But this is precisely the definition of the centers of $T \setminus L$. As a consequence, we see that the centers of T are precisely the centers of $T \setminus L$. This proves Lemma 5.5.10 **(f)**. ☐

Proof of Theorem 5.5.9. We shall prove parts **(a)** and **(b)** of Theorem 5.5.9 by strong induction on $|\mathrm{V}(T)|$:

Induction step: Consider a tree T. Assume that parts **(a)** and **(b)** of Theorem 5.5.9 are true for any tree with fewer than $|\mathrm{V}(T)|$ many vertices. We must now prove these parts for our tree T.

If $|\mathrm{V}(T)| \leq 2$, then both parts are obvious. Hence, WLOG assume that $|\mathrm{V}(T)| > 2$. Thus, the tree T has more than 2 vertices. Let L be the set of all leaves of T. Note that $|L| \geq 2$ (since we know that any tree with at least 2 vertices has at least 2 leaves). Define the multigraph $T \setminus L$ as in Lemma 5.5.10. Then, Lemma 5.5.10 **(f)** shows that the centers of T are precisely the centers of $T \setminus L$.

However, Lemma 5.5.10 **(a)** yields that $T \setminus L$ is again a tree. This tree has fewer vertices than T (since $|L| \geq 2 > 0$). Hence, by the induction hypothesis, both parts **(a)** and **(b)** of Theorem 5.5.9 are true for the tree $T \setminus L$ instead of T.

In other words, the tree $T \setminus L$ has either 1 or 2 centers, and if it has 2 centers, then these 2 centers are adjacent. Since the centers of T are precisely the centers of $T \setminus L$, we can rewrite this as follows: The tree T has either 1 or 2 centers, and if it has 2 centers, then these 2 centers are adjacent. In other words, parts **(a)** and **(b)** of Theorem 5.5.9 hold for our tree T. This completes the induction step. Thus, parts **(a)** and **(b)** of Theorem 5.5.9 are proved.

(c) This follows from Lemma 5.5.10 **(f)**. Indeed, if T has at most 2 vertices, then all vertices of T are centers of T (this is trivial to check). If not, then each "leaf-removal" step of our algorithm leaves the set of centers of T unchanged (by Lemma 5.5.10 **(f)**), and thus the centers of the original tree T are precisely the centers of the tree that remains at the end of the algorithm. But the latter tree has at most 2 vertices, and thus its centers are precisely its vertices. So the centers of T are precisely the vertices that remain at the end of the algorithm. Theorem 5.5.9 **(c)** is proven. ☐

The following exercise shows another approach to the centers of a tree:

Exercise 5.14. Let T be a tree. Let $\mathbf{p} = (p_0, *, p_1, *, p_2, \ldots, *, p_m)$ be a longest path of T. (We write asterisks for the edges since we don't need to name them.)

Prove the following:

(a) If m is even, then the only center of T is $p_{m/2}$.

(b) If m is odd, then the two centers of T are $p_{(m-1)/2}$ and $p_{(m+1)/2}$.

Remark 5.5.12. Exercise 5.14 is a result by Arthur Cayley from 1875. It shows once again that each tree has exactly one center or two adjacent centers, and also shows that any two longest paths of a tree have a common vertex.

The notion of a **centroid** of a tree is a relative of the notion of a center. We briefly discuss it in the following exercise:

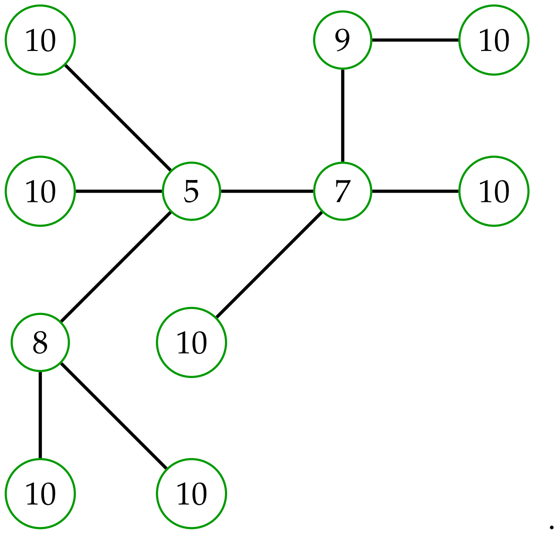
Exercise 5.15. Let T be a tree. For any vertex v of T, we let c_v denote the size of the largest component of the graph $T \setminus v$. (Recall that $T \setminus v$ is the graph obtained from T by removing the vertex v and all edges that contain v. Note that a component (according to our definition) is a set of vertices; thus, its size is the number of vertices in it.)

The vertices v of T that minimize the number c_v are called the **centroids** of T.

(a) Prove that T has no more than two centroids, and furthermore, if T has two centroids, then these two centroids are adjacent.

(b) Find a tree T such that the centroid(s) of T are distinct from the center(s) of T.

[**Example:** Here is an example of a tree T, where each vertex v is labelled with the corresponding number c_v:



Thus, the vertex labelled 5 is the only centroid of this tree T.]

Note the analogy between Exercise 5.15 **(a)** and Theorem 5.5.9 **(a)** and **(b)**.

5.6. Arborescences

5.6.1. Definitions

Enough about undirected graphs.

What would be a directed analogue of a tree? I.e., what kind of digraphs play the same role among digraphs that trees do among undirected graphs?

Trees are graphs that are connected and have no cycles. This suggests two directed versions:

- We can study digraphs that are strongly connected and have no cycles. Unfortunately, there is not much to study: Any such digraph has only 1 vertex and no arcs. (Make sure you understand why!)

- We can drop the connectedness requirement. Digraphs that have no cycles are called **acyclic**, and more typically they are called **dags** (short for "directed acyclic graphs").

  However, these dags aren't quite like trees. For example, a tree always has fewer edges than vertices, but a dag can have more arcs than vertices.[39]

Here is a more convincing analogue of trees for digraphs:[40]

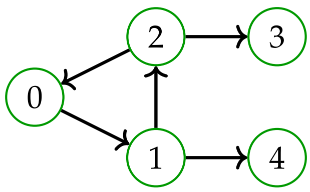
Definition 5.6.1. Let D be a multidigraph. Let r be a vertex of D.

(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 D is an **arborescence rooted from** r if r is a from-root of D and the undirected multigraph D^{und} has no cycles. (Recall that D^{und} is the multigraph obtained from D by turning each arc into an undirected edge. Parallel arcs are not merged into one!)

Of course, there are analogous notions of a "to-root" and an "arborescence rooted towards r", but these are just the same notions that we just defined with all arrows reversed. So we need not study them separately; we can just take any property of "rooted from" and reverse all arcs to make it into a property of "rooted to".

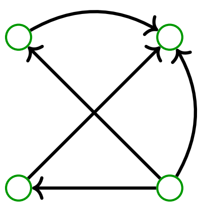
Example 5.6.2. The multidigraph



has three from-roots (namely, 0, 1 and 2). It is not an arborescence rooted from any of them, because turning each arc into an undirected edge yields a graph with a cycle.

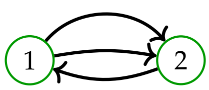
---

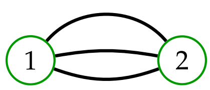
[39]For example, here is a dag with 4 vertices and 5 arcs:



.

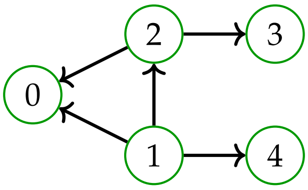
[40]We recall that we defined a multigraph D^{und} for every multidigraph D (in Definition 4.4.1). Roughly speaking, this multigraph D^{und} is obtained by "forgetting the directions" of the arcs of D. Parallel arcs are not merged into one. For example,

if $D =$ 

, then $D^{\text{und}} =$ 

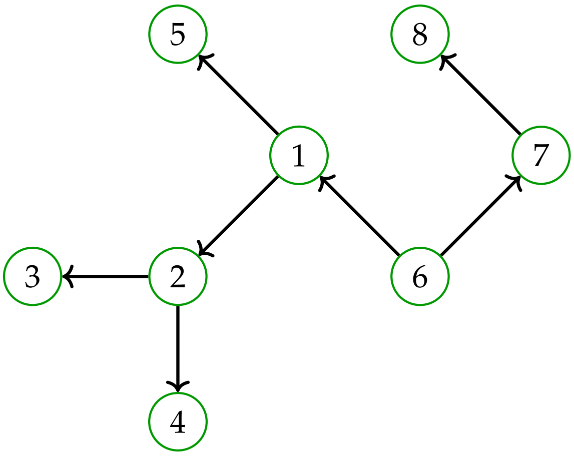
.

If we reverse the arc from 0 to 1, then we obtain a multidigraph



which has only one from-root (namely, 1) and is still not an arborescence (for the same reason as before).

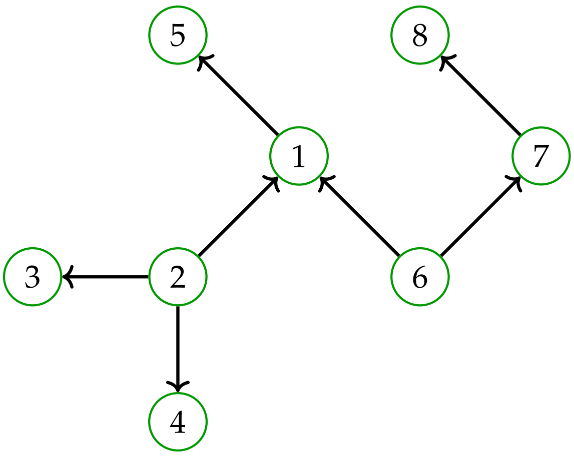
Example 5.6.3. Consider the following multidigraph:



.

This is an arborescence rooted from 6. Indeed, it has paths from 6 to all vertices, and turning each arc into an undirected edge yields a tree.

If we reverse the arc from 1 to 2, we obtain a multidigraph



,

which is **not** an arborescence, because it has no from-root anymore.

5.6.2. Arborescences vs. trees: statement

The above examples suggest that an arborescence rooted from r is basically the same as a tree, whose all edges have been "oriented away from r". More precisely:

Theorem 5.6.4. 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}).

This is an easy theorem to believe, but an annoyingly hard one to formally prove in full detail! We shall prove this theorem later.

5.6.3. The arborescence equivalence theorem

First, let us show another bunch of equivalent criteria for arborescences, imitating the tree equivalence theorem (Theorem 5.2.4):

Theorem 5.6.5 (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 multigraph 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 the vertex r will no longer be a from-root of the resulting multidigraph.
- **Statement A6:** We have $\deg^- r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$.

Proof. We will prove the implications A1$\Longrightarrow$A4$\Longrightarrow$A5$\Longrightarrow$A6$\Longrightarrow$A2$\Longrightarrow$A3$\Longrightarrow$A1. Since these implications form a cycle that includes all six statements, this will entail that all six statements are equivalent.

Before we prove these implications, we introduce a notation: If a is any arc of D, then $D \setminus a$ shall denote the multidigraph obtained from D by removing this arc a. (Formally, this means that $D \setminus a := \left(V,\ A \setminus \{a\}\ ,\ \psi\mid_{A \setminus \{a\}}\right)$.)

We now come to the proofs of the promised implications.

Proof of the implication A1$\Longrightarrow$A4: Assume that Statement A1 holds. Thus, D is an arborescence rooted from r. In other words, r is a from-root of D and the undirected multigraph D^{und} has no cycles.

We must show that for each vertex $v \in V$, the multidigraph D has a unique walk from r to v. The existence of such a walk is clear (because r is a from-root of D). It is the uniqueness that we need to prove.

Assume the contrary. Thus, there exists a vertex $v \in V$ such that two distinct walks $\mathbf{u}$ and $\mathbf{v}$ from r to v exist. However, the multigraph D has no loops (since any loop of D would be a loop of D^{und}, and thus create a cycle of D^{und}, but we know that D^{und} has no cycles). Hence, any walk of D is automatically a backtrack-free walk of D^{und} (indeed, it is backtrack-free because the only way two consecutive arcs of a walk in a **digraph** can be equal is if they are loops). Therefore, the two walks $\mathbf{u}$ and $\mathbf{v}$ of D are two backtrack-free walks of D^{und}. Thus, there are two distinct backtrack-free walks from r to v in D^{und} (namely, $\mathbf{u}$ and $\mathbf{v}$). Theorem 5.1.3 thus lets us conclude that D^{und} has a cycle. But this contradicts the fact that D^{und} has no cycles.

This contradiction shows that our assumption was wrong. Hence, we have proved that for each vertex $v \in V$, the multidigraph D has a unique walk from r to v. In other words, Statement A4 holds.

Proof of the implication A4$\Longrightarrow$A5: Assume that Statement A4 holds.

Let now a be any arc of D. We shall show that r is not a from-root of the multidigraph $D \setminus a$.

Indeed, let s be the source and t the target of the arc a. We shall show that the digraph $D \setminus a$ has no path from r to t.

Indeed, assume the contrary. Thus, $D \setminus a$ has some path $\mathbf{p}$ from r to t. This path does not use the arc a (since it is a path of $D \setminus a$).

On the other hand, we have assumed that Statement A4 holds. Applying this statement to $v = s$, we conclude that the multidigraph D has a unique walk from r to s. Let $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ be this walk. By appending the arc a and the vertex t to its end, we extend it to a longer walk

$$(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k, a, t)\,,$$

which is a walk from r to t. We denote this walk by $\mathbf{q}$.

We have now found two walks from r to t in the digraph D: namely, the path $\mathbf{p}$ and the walk $\mathbf{q}$. These two walks are distinct (since $\mathbf{q}$ uses the arc a, but $\mathbf{p}$ does not). However, Statement A4 (applied to $v = t$) yields that the multidigraph D has a **unique** walk from r to t. This contradicts the fact that we just have found two distinct such walks.

This contradiction shows that our assumption was false. Hence, the digraph $D \setminus a$ has no path from r to t. Thus, r is not a from-root of $D \setminus a$.

Forget that we fixed a. We have now proved that if a is any arc of D, then r is not a from-root of $D \setminus a$. In other words, if we remove any arc from D, then the vertex r will no longer be a from-root of the resulting multidigraph. Thus, Statement A5 holds.

Proof of the implication A5$\Longrightarrow$A6: Assume that Statement A5 holds. We must prove that Statement A6 holds. In other words, we must prove that $\deg^- r = 0$, and that each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$.

Let us first prove that $\deg^- r = 0$. Indeed, assume the contrary. Thus, $\deg^- r \neq 0$, so that there exists an arc a with target r. We shall show that r is a from-root of $D \setminus a$.

The arc a has target r. Thus, a path that starts at r cannot use this arc a (because this arc would lead it back to r, but a path is not allowed to revisit any vertex), and therefore must be a path of $D \setminus a$. Thus we have shown that any path of D that starts at r is also a path of $D \setminus a$. However, for each vertex v of D, the digraph D has a path from r to v (since r is a from-root of D). This path is also a path of $D \setminus a$ (since any path of D that starts at r is also a path of $D \setminus a$). Thus, for each vertex v of $D \setminus a$, the digraph $D \setminus a$ has a path from r to v. In other words, r is a from-root of $D \setminus a$. However, we have assumed that Statement A5 holds. Thus, in particular, if we remove the arc a from D, then the vertex r will no longer be a from-root of the resulting multidigraph. In other words, r is not a from-root of $D \setminus a$. But this contradicts the fact that r is a from-root of $D \setminus a$.

This contradiction shows that our assumption was false. Hence, $\deg^- r = 0$ is proved.

Now, let $v \in V \setminus \{r\}$ be arbitrary. We must show that $\deg^- v = 1$.

Indeed, assume the contrary. Thus, $\deg^- v \neq 1$. Using the fact that r is a from-root of D, it is thus easy to see that $\deg^- v \geq 2$ [41]. Hence, there exist two distinct arcs a and b with target v. Consider these arcs a and b.

We are in one of the following three cases:

Case 1: The digraph $D \setminus a$ has a path from r to v.

Case 2: The digraph $D \setminus b$ has a path from r to v.

Case 3: Neither the digraph $D \setminus a$ nor the digraph $D \setminus b$ has a path from r to v.

Let us first consider Case 1. In this case, the digraph $D \setminus a$ has a path from r to v. Let $\mathbf{p}$ be such a path.

We have assumed that Statement A5 holds. Thus, in particular, if we remove the arc a from D, then the vertex r will no longer be a from-root of the resulting

[41] *Proof.* Since r is a from-root of D, we know that the digraph D has a path from r to v. Since $v \neq r$ (because $v \in V \setminus \{r\}$), this path must have at least one arc. The last arc of this path is clearly an arc with target v. Thus, there exists at least one arc with target v. In other words, $\deg^- v \geq 1$. Combining this with $\deg^- v \neq 1$, we obtain $\deg^- v > 1$. In other words, $\deg^- v \geq 2$.

multidigraph. In other words, r is not a from-root of $D \setminus a$. In other words, there exists a vertex $w \in V$ such that the digraph $D \setminus a$ has no path from r to w (by the definition of a "from-root"). Consider this vertex w.

The digraph D has a path $\mathbf{q}$ from r to w (since r is a from-root of D). Consider this path $\mathbf{q}$. If the path $\mathbf{q}$ did not use the arc a, then it would be a path of $D \setminus a$ as well, but this would contradict the fact that $D \setminus a$ has no path from r to w. Thus, the path $\mathbf{q}$ must use the arc a.

Consider the part of $\mathbf{q}$ that comes after the arc a. This part must be a path from v to w (since the arc a has target v, whereas the path $\mathbf{q}$ has ending point w). Let us denote this path by $\mathbf{q}'$. Thus, the path $\mathbf{q}'$ does not use the arc a (since it was defined as the part of $\mathbf{q}$ that comes after a). Hence, $\mathbf{q}'$ is a path of $D \setminus a$.

Now, we know that the digraph $D \setminus a$ has a path $\mathbf{p}$ from r to v as well as a path $\mathbf{q}'$ from v to w. Splicing these paths together, we obtain a walk $\mathbf{p} * \mathbf{q}'$ from r to w. So we know that $D \setminus a$ has a walk from r to w. According to Corollary 3.3.10, we thus conclude that $D \setminus a$ has a path from r to w. This contradicts the fact that $D \setminus a$ has no path from r to w.

We have thus obtained a contradiction in Case 1.

The same argument (but with the roles of a and b interchanged) results in a contradiction in Case 2.

Let us finally consider Case 3. In this case, neither the digraph $D \setminus a$ nor the digraph $D \setminus b$ has a path from r to v. However, the digraph D has a path $\mathbf{p}$ from r to v (since r is a from-root of D). Consider this path $\mathbf{p}$. If this path $\mathbf{p}$ did not use the arc a, then it would be a path of $D \setminus a$, but this would contradict our assumption that the digraph $D \setminus a$ has no path from r to v. Thus, this path $\mathbf{p}$ must use the arc a. For a similar reason, it must also use the arc b. However, the two arcs a and b have the same target (viz., v) and thus cannot both appear in the same path (since a path cannot visit a vertex more than once). This contradicts the fact that the path $\mathbf{p}$ uses both arcs a and b. Hence, we have found a contradiction in Case 3.

We have now found contradictions in all three Cases 1, 2 and 3. This contradiction shows that our assumption was false. Hence, $\deg^- v = 1$ is proved.

We have now proved that each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$. Since we have also shown that $\deg^- r = 0$, we thus have proved Statement A6.

Proof of the implication A6⟹A2: Assume that Statement A6 holds. We must prove that Statement A2 holds. However, Proposition 4.2.3 yields

$$\begin{aligned} |A| = \sum_{v \in V} \deg^- v &= \underbrace{\deg^- r}_{\substack{=0 \\ \text{(by Statement A6)}}} + \sum_{v \in V \setminus \{r\}} \underbrace{\deg^- v}_{\substack{=1 \\ \text{(by Statement A6)}}} \\ &= 0 + \sum_{v \in V \setminus \{r\}} 1 = \sum_{v \in V \setminus \{r\}} 1 = |V \setminus \{r\}| = |V| - 1. \end{aligned}$$

Hence, Statement A2 holds.

Proof of the implication A2⟹A3: Assume that Statement A2 holds. We must prove that Statement A3 holds.

For each $v \in V$, the digraph D has a path from r to v (since r is a from-root of D). Thus, for each $v \in V$, the graph D^{und} has a path from r to v (since any path of D is a path of D^{und}). Therefore, any two vertices u and v of D^{und} are path-connected in D^{und} (because we can get from u to v via r, according to the previous sentence). Therefore, the graph D^{und} is connected (since it has at least one vertex[42]). Moreover, its number of edges is $|A| = |V| - 1$ (by Statement A2). Therefore, the multigraph D^{und} satisfies the Statement T4 of the tree equivalence theorem (Theorem 5.2.4). Consequently, it satisfies Statement T1 of that theorem as well. In other words, it is a tree. This proves Statement A3.

Proof of the implication A3$\Longrightarrow$A1: Assume that Statement A3 holds. We must prove that Statement A1 holds.

The multigraph D^{und} is a tree (by Statement A3), and thus is a forest; hence, it has no cycles. Since we also know that r is a from-root of D, we thus conclude that D is an arborescence rooted from r (by the definition of an arborescence). In other words, Statement A1 is satisfied.

We have now proved all six implications in the chain A1$\Longrightarrow$A4$\Longrightarrow$A5$\Longrightarrow$A6$\Longrightarrow$A2$\Longrightarrow$A3$\Longrightarrow$A1. Thus, all six statements A1, A2, ..., A6 are equivalent. This proves Theorem 5.6.5. $\square$

Exercise 5.16. Let $D = (V, A, \phi)$ be a multidigraph that has no cycles[43]. Let $r \in V$ be some vertex of D. Prove the following:

(a) If $\deg^- u > 0$ holds for all $u \in V \setminus \{r\}$, then r is a from-root of D.

(b) If $\deg^- u = 1$ holds for all $u \in V \setminus \{r\}$, then D is an arborescence rooted from r.

5.7. Arborescences vs. trees

Our next goal is to prove Theorem 5.6.4, which connects arborescences with trees.

To prove it formally, we introduce a few notations regarding trees. First, we recall the notion of a distance (Definition 5.5.1). We claim the following simple property of distances in trees:

Proposition 5.7.1. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Let e be an edge of T, and let u and v be its two endpoints. Then, the distances $d(r, u)$ and $d(r, v)$ differ by exactly 1 (that is, we have either $d(r, u) = d(r, v) + 1$ or $d(r, v) = d(r, u) + 1$).

[42] This is because $r \in V$.

[43] Recall that cycles in a digraph have to be directed cycles – i.e., each arc is traversed from its source to its target.

Proof. We recall that since T is a tree, the distance $d(p,q)$ between two vertices p and q of T is simply the length of the path from p to q. (This path is unique, since T is a tree.)

Let $\mathbf{p}$ be the path from r to u. Then, we are in one of the following two cases:

Case 1: The edge e is an edge of $\mathbf{p}$.

Case 2: The edge e is not an edge of $\mathbf{p}$.

Consider Case 1. In this case, e must be the **last** edge of $\mathbf{p}$ (since otherwise, $\mathbf{p}$ would visit u more than once, but $\mathbf{p}$ cannot do this, since $\mathbf{p}$ is a path). Thus, if we remove this last edge e (and the vertex u) from $\mathbf{p}$, then we obtain a path from r to v. This path is exactly one edge shorter than $\mathbf{p}$. Thus, $d(r,v) = d(r,u) - 1$, so that $d(r,u) = d(r,v) + 1$. So we are done in Case 1.

Now, consider Case 2. In this case, the edge e is not an edge of $\mathbf{p}$. Thus, we can append e and v to the end of the path $\mathbf{p}$, and the result will be a backtrack-free walk $\mathbf{p}'$. However, a backtrack-free walk in a tree is always a path (since otherwise, it would contain a cycle[44], but a tree has no cycles). Thus, $\mathbf{p}'$ is a path from r to v, and it is exactly one edge longer than $\mathbf{p}$ (by its construction). Therefore, $d(r,v) = d(r,u) + 1$. So we are done in Case 2.

Now, we are done in both cases, so that Proposition 5.7.1 is proven. $\square$

Definition 5.7.2. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Let e be an edge of T. By Proposition 5.7.1, 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.

- **(a)** We define the r-**parent** of e to be the endpoint of e whose distance to r is the smallest. We denote this endpoint by e^{-r}.
- **(b)** We define the r-**child** of e to be the endpoint of e whose distance to r is the largest. We denote this endpoint by e^{+r}.

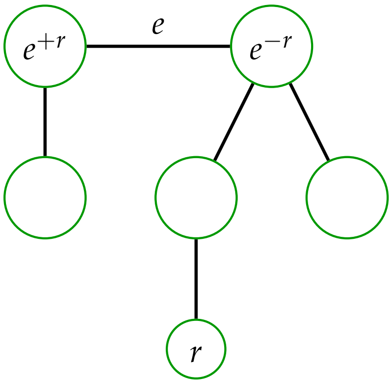
Thus, by Proposition 5.7.1, we have

$$d\left(r, e^{+r}\right) = d\left(r, e^{-r}\right) + 1.$$

Example 5.7.3. Here is a tree T, a vertex r, an edge e and its r-parent e^{-r} and

[44] by Proposition 5.1.2

its r-child e^{+r}:

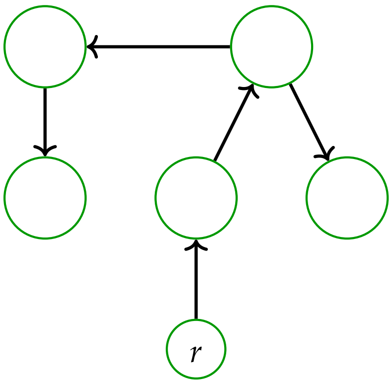


Definition 5.7.4. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Then, we define a multidigraph $T^{r\rightarrow}$ by

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

where $\psi : E \rightarrow V \times V$ is the map that sends each edge $e \in E$ to the pair (e^{-r}, e^{+r}). Colloquially speaking, this means that $T^{r\rightarrow}$ is the multidigraph obtained from T by turning each edge e into an arc from its r-parent e^{-r} to its r-child e^{+r}. This is what we mean when we speak of "orienting each edge of T away from r" in Theorem 5.6.4.

Example 5.7.5. If T is the tree from Example 5.7.3, then $T^{r\rightarrow}$ is the following multidigraph:



Now, Theorem 5.6.4 can be rewritten as follows:

Theorem 5.7.6. 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 we have $D = \left(D^{\text{und}}\right)^{r\rightarrow}$. (This is a honest equality, not just some isomorphism.)

The proof of this theorem is best organized by splitting into two lemmas:

Lemma 5.7.7. Let $T = (V, E, \varphi)$ be a tree. Let $r \in V$ be a vertex of T. Then, the multidigraph $T^{r\rightarrow}$ is an arborescence rooted from r.

Proof. The idea is to show that if $\mathbf{p}$ is a path from r to some vertex v in the tree T, then $\mathbf{p}$ is also a path in the digraph $T^{r\rightarrow}$, because all the edges of $\mathbf{p}$ have been "oriented correctly" (i.e., their orientation matches how they are used in $\mathbf{p}$).

Here are the details: Clearly, $(T^{r\rightarrow})^{\text{und}} = T$. Hence, the graph $(T^{r\rightarrow})^{\text{und}}$ is a tree and hence has no cycles. Thus, it suffices to prove that r is a from-root of $T^{r\rightarrow}$. In other words, we must prove that

$$T^{r\rightarrow} \text{ has a path from } r \text{ to } v \tag{20}$$

for each $v \in V$.

We shall prove (20) by induction on $d(r, v)$ (where d means the distance on the tree T):

Base case: If $v \in V$ satisfies $d(r, v) = 0$, then $v = r$, and thus $T^{r\rightarrow}$ has a path from r to v (namely, the trivial path (r)). Thus, (20) is proved for $d(r, v) = 0$.

Induction step: Let $k \in \mathbb{N}$. Assume (as the induction hypothesis) that (20) holds for each $v \in V$ satisfying $d(r, v) = k$. We must now prove the same for each $v \in V$ satisfying $d(r, v) = k + 1$.

So let $v \in V$ satisfy $d(r, v) = k + 1$. Then, the path of T from r to v has length $k + 1$. Let $\mathbf{p}$ be this path, let e be its last edge, and let u be its second-to-last vertex (so that its last edge e has endpoints u and v). Then, by removing the last edge e from the path $\mathbf{p}$, we obtain a path from r to u that is one edge shorter than $\mathbf{p}$. Hence, $d(r, u) = d(r, v) - 1 < d(r, v)$. Consequently, the edge e has r-parent u and r-child v (by Definition 5.7.2). In other words, $e^{-r} = u$ and $e^{+r} = v$. Therefore, in the digraph $T^{r\rightarrow}$, the edge e is an arc from u to v (by Definition 5.7.4). Moreover, we have $d(r, u) = d(r, v) - 1 = k$ (since $d(r, v) = k + 1$); therefore, the induction hypothesis tells us that (20) holds for u instead of v. In other words, $T^{r\rightarrow}$ has a path from r to u. Attaching the arc e and the vertex v to this path, we obtain a walk of $T^{r\rightarrow}$ from r to v (since e is an arc from u to v in $T^{r\rightarrow}$). Thus, the digraph $T^{r\rightarrow}$ has a walk from r to v, therefore also a path from r to v. Hence, (20) holds for our v. This completes the induction step.

Thus, (20) is proved by induction. As we explained above, this yields Lemma 5.7.7. $\square$

Lemma 5.7.8. Let $D = (V, A, \psi)$ be an arborescence rooted from r (for some $r \in V$). Let $a \in A$ be an arc of D. Let s be the source of a, and let t be the target of a. Then:

(a) We have $d(r, s) < d(r, t)$, where d means distance on the tree D^{und}.

(b) In the multidigraph $\left(D^{\text{und}}\right)^{r\rightarrow}$, the arc a has source s and target t.

Proof. **(a)** The vertex r is a from-root of D (since D is an arborescence rooted from r). Thus, D has a path from r to t. Let $\mathbf{p}$ be this path. Note that $\deg^- t \geq 1$, since t is the target of at least one arc (namely, of a).

The digraph D is an arborescence rooted from r, and thus satisfies Statement A6 in the arborescence equivalence theorem (Theorem 5.6.5). In other words, we have

$$\deg^- r = 0 \qquad \text{and} \qquad \deg^- v = 1 \text{ for each } v \in V \setminus \{r\}.$$

In particular, this entails $\deg^- v \leq 1$ for each $v \in V$. Applying this to $v = t$, we obtain $\deg^- t \leq 1$. Hence, the arc a is the **only** arc whose target is t.

We have $t \neq r$ (since $\deg^- r = 0$ but $\deg^- t \geq 1 > 0$). Thus, the path $\mathbf{p}$ from r to t has at least one arc. Its last arc is therefore an arc whose target is t. Hence, this last arc is a (since a is the **only** arc whose target is t).

If we remove this last arc from the path $\mathbf{p}$, then we obtain a path $\mathbf{p}'$ from r to s (since s is the source of a).

However, each path of D is a path of D^{und}. Thus, in particular, $\mathbf{p}$ is a path of D^{und} from r to t, while $\mathbf{p}'$ is a path of D^{und} from r to s. Since $\mathbf{p}'$ is exactly one edge shorter than $\mathbf{p}$, we thus obtain $d(r,s) = d(r,t) - 1 < d(r,t)$. This proves Lemma 5.7.8 **(a)**.

(b) The arc a of the digraph D has source s and target t. Hence, the edge a of the tree D^{und} has endpoints s and t. Since $d(r,s) < d(r,t)$ (by part **(a)**), this entails that its r-parent is s and its r-child is t (by Definition 5.7.2). Thus, in the digraph $\left(D^{\text{und}}\right)^{r\to}$, this edge a becomes an arc with source s and target t (by Definition 5.7.4). This proves Lemma 5.7.8 **(b)**. $\square$

Proof of Theorem 5.7.6. If (V, A, ψ) is a multidigraph, then we shall refer to the map $\psi : A \to V \times V$ (which determines the source and the target of each arc) as the "psi-map" of this multidigraph.

Write the multidigraph D as $D = (V, A, \psi)$. We shall now prove the implications C1$\Longrightarrow$C2 and C2$\Longrightarrow$C1 separately:

Proof of the implication C1$\Longrightarrow$C2: Assume that Statement C1 holds. That is, D is an arborescence rooted from r. We must prove Statement C2. In other words, we must prove that the undirected multigraph D^{und} is a tree, and that $D = \left(D^{\text{und}}\right)^{r\to}$.

It is clear (by the definition of an arborescence) that D^{und} is a tree. It thus remains to prove that $D = \left(D^{\text{und}}\right)^{r\to}$.

The multidigraphs D and $\left(D^{\text{und}}\right)^{r\to}$ have the same set of vertices (namely, V) and the same set of arcs (namely, A); we therefore just need to show that their psi-maps are the same. In other words, we need to show that $\psi' = \psi$, where ψ' is the psi-map of $\left(D^{\text{und}}\right)^{r\to}$.

Let $a \in A$ be arbitrary. Let $\psi(a) = (s,t)$. Thus, the arc a of D has source s and target t. Lemma 5.7.8 **(b)** therefore shows that in the multidigraph $\left(D^{\text{und}}\right)^{r\to}$, the arc a has source s and target t as well. In other words, $\psi'(a) = (s,t)$ (since ψ' is the psi-map of this multidigraph). Hence, $\psi'(a) = (s,t) = \psi(a)$.

Forget that we fixed a. We thus have shown that $\psi'(a) = \psi(a)$ for each $a \in A$. In other words, $\psi' = \psi$. As explained above, this completes the proof of Statement C2.

Proof of the implication C2$\Longrightarrow$C1: Assume that Statement C2 holds. Thus, the undirected multigraph D^{und} is a tree, and we have $D = \left(D^{\text{und}}\right)^{r\to}$. Hence, Lemma 5.7.7 (applied to $T = D^{\text{und}}$) yields that the multidigraph $\left(D^{\text{und}}\right)^{r\to}$ is an arborescence rooted from r. In other words, D is an arborescence rooted from r (since $D = \left(D^{\text{und}}\right)^{r\to}$). This shows that Statement C1 holds.

Having now proved both implications C1$\Longrightarrow$C2 and C2$\Longrightarrow$C1, we conclude that Statements C1 and C2 are equivalent. Thus, Theorem 5.7.6 is proved. $\square$

Oof.

Let's get one more consequence out of this. First, let us show that an arborescence can have only one root:

Proposition 5.7.9. Let D be an arborescence rooted from r. Then, r is the **only** root of D.

Proof of Proposition 5.7.9. Assume the contrary. Thus, D has another root s distinct from r. Hence, D has a path from r to s (since r is a root) as well as a path from s to r (since s is a root). Combining these paths gives a circuit of length > 0. However, a circuit of length > 0 in a digraph must always contain a cycle (since Proposition 4.5.9 shows that it either is a path or contains a cycle; but it clearly cannot be a path). Hence, D has a cycle. Therefore, D^{und} also has a cycle (since any cycle of D is a cycle of D^{und}). However, D^{und} has no cycles (since D is an arborescence rooted from r). The preceding two sentences contradict each other. This shows that the assumption was wrong, and Proposition 5.7.9 is proven. $\square$

Definition 5.7.10. 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, this r is uniquely determined as the only root of D (by Proposition 5.7.9).

Theorem 5.7.11. There are two mutually inverse maps

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

and

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

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

Proof. The map

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

is well-defined because of Lemma 5.7.7. The map

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

is well-defined because if D is an arborescence, then D^{und} is a tree. In order to show that these two maps are mutually inverse, we must check the following two statements:

1. Each arborescence D satisfies $\left(D^{\text{und}}\right)^{r\to} = D$, where r is the root of D;

2. Each pair (T,r) of a tree T and a vertex r of T satisfies $\left(T^{r\to}\right)^{\text{und}} = T$ and $\sqrt{\left(T^{r\to}\right)^{\text{und}}} = r$.

However, Statement 1 follows from Theorem 5.7.6 (specifically, from the implication C1$\Longrightarrow$C2 in Theorem 5.7.6). Statement 2 follows from Lemma 5.7.7 (more precisely, the $\left(T^{r\to}\right)^{\text{und}} = T$ part of Statement 2 is obvious, whereas the $\sqrt{\left(T^{r\to}\right)^{\text{und}}} = r$ part follows from Lemma 5.7.7). Thus, Theorem 5.7.11 is proved. $\square$

Theorem 5.7.11 formalizes the idea that an arborescence is "just a tree with a chosen vertex". For this reason, arborescences are sometimes called "oriented trees", but this name is also shared with a more general notion, which is why I avoid it.

Exercise 5.17. Let $G = (V, E, \varphi)$ be a connected multigraph such that $|E| \geq |V|$. Show that there exists an injective map $f : V \to E$ such that for each vertex $v \in V$, the edge $f(v)$ contains v.

(In other words, show that we can assign to each vertex an edge that contains this vertex in such a way that no edge is assigned twice.)

5.8. Spanning arborescences

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

Definition 5.8.1. 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.

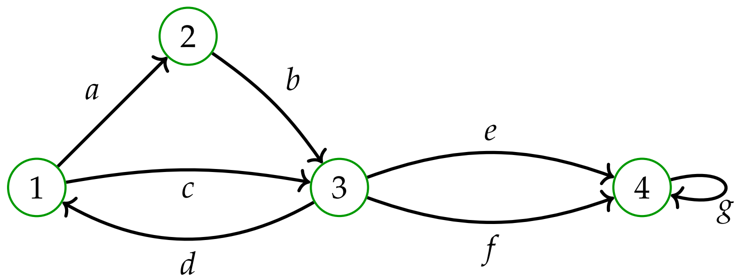
In other words, it means a submultidigraph of D with the same vertex set as D.

In other words, it means a multidigraph obtained from D by removing some arcs, but leaving all vertices untouched.

Definition 5.8.2. 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.

Example 5.8.3. Let $D = (V, A, \psi)$ be the following multidigraph:

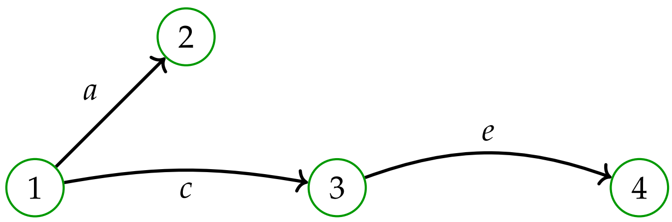
$D =$



.

Is there a spanning arborescence of D rooted from 1 ? Yes, for instance,

$\left(V,\ \{a,c,e\}\ ,\ \psi\mid_{\{a,c,e\}}\right) =$



.

By abuse of notation, we shall refer to this spanning arborescence simply as $\{a, c, e\}$ (since a spanning subdigraph of D is uniquely determined by its arc set). Another spanning arborescence of D rooted from 1 is $\{a, b, e\}$. Yet another is $\{a, b, f\}$. A non-example is $\{a, d, f\}$ (indeed, this is an arborescence rooted from 3, not from 1).

Is there a spanning arborescence of D rooted from 2 ? Yes, for example $\{b, d, f\}$.

Is there a spanning arborescence of D rooted from 4 ? No, since 4 is not a from-root of D.

This illustrates a first obstruction to the existence of spanning arborescences: Namely, a digraph D can have a spanning arborescence rooted from r only if r is a from-root. This necessary criterion is also sufficient:

Theorem 5.8.4. Let D be a multidigraph. Let r be a from-root of D. Then, D has a spanning arborescence rooted from r.

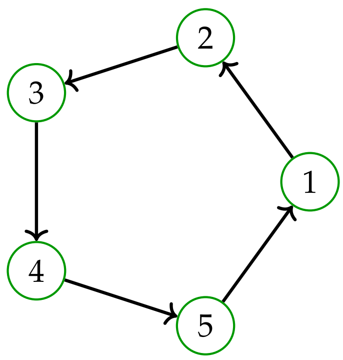
Proof. This is an analogue of the "every connected multigraph has a spanning tree" theorem (Theorem 5.4.6) that we proved in 4 ways. At least the first proof easily adapts to the directed case:

Remove arcs from D one by one, but in such a way that the "rootness of r" (that is, the property that r is a root of our multidigraph) is preserved. So we can only remove an arc if r remains a root afterwards.

Clearly, this removing process will eventually come to an end, since D has only finitely many arcs. Let D' be the multidigraph obtained at the end of this process. Then, r is still a root of D', but we cannot remove any more arcs from D' without breaking the rootness of r. That is, if we remove any arc from D', then the vertex r will no longer be a from-root of the resulting multidigraph. This means that D' satisfies Statement A5 from the arborescence equivalence theorem (Theorem 5.6.5). Thus, D' satisfies Statement A1 as well (since all six statements A1, A2, ..., A6 are equivalent). In other words, D' is an arborescence rooted from r. Since D' is a spanning subdigraph of D, we thus conclude that D has a spanning arborescence rooted from r (namely, D'). This proves Theorem 5.8.4. $\square$

Question 5.8.5. Can the other three proofs of Theorem 5.4.6 be adapted to Theorem 5.8.4, too?

Example 5.8.6. Let n be a positive integer. The ***n*-cycle digraph** $\overrightarrow{C}_n$ is defined to be the simple digraph with vertices $1, 2, \ldots, n$ and arcs $12,\ 23,\ 34,\ \ldots,\ (n-1)\,n,\ n1$. (Here is how it looks for $n = 5$:



)

Note that this digraph $\overrightarrow{C}_n$ is a directed analogue of the cycle graph C_n. As we recall from Example 5.4.4, the cycle graph C_n has n spanning trees.

In contrast, the digraph $\overrightarrow{C}_n$ has only one spanning arborescence rooted from 1. This spanning arborescence is the subdigraph of $\overrightarrow{C}_n$ obtained by removing the arc $n1$.

Proof. If we remove the arc $n1$ from $\overrightarrow{C}_n$, then we obtain the simple digraph E with vertices $1, 2, \ldots, n$ and arcs $12, 23, \ldots, (n-1)\,n$. This digraph E is easily seen to be an arborescence rooted from 1 (indeed, 1 is a from-root of E, and the underlying undirected graph $E^{\text{und}} = P_n$ has no cycles). Thus, E is a spanning arborescence of $\overrightarrow{C}_n$ rooted from 1.

We shall now prove that it is the only such arborescence. Indeed, let F be any spanning arborescence of $\overrightarrow{C}_n$ rooted from 1. Then, 1 is a from-root of F. Hence, for each vertex $v \in \{2, 3, \ldots, n\}$, the digraph F must have a path from 1 to v, and thus must contain an arc with target v (namely, the last arc of this path). This arc must be $(v-1, v)$ (since this is the only arc of $\overrightarrow{C}_n$ with target v). Thus, for each vertex $v \in \{2, 3, \ldots, n\}$, the digraph F must contain the arc $(v-1, v)$. In other words, the digraph F must contain all $n-1$ arcs $12, 23, \ldots, (n-1)\,n$. If F were to also contain the remaining arc $n1$ of $\overrightarrow{C}_n$, then the underlying undirected graph $F^{\text{und}} = C_n$ would contain a cycle, which would contradict F being an arborescence. Hence, F cannot contain the arc $n1$. Thus, F contains the $n-1$ arcs $12, 23, \ldots, (n-1)\,n$ and no others. In other words, $F = E$. This shows that any spanning arborescence of $\overrightarrow{C}_n$ rooted from 1 must be E. In other words, E is the only spanning arborescence of $\overrightarrow{C}_n$ rooted from 1. This completes the proof of Example 5.8.6. ☐

5.9. The BEST theorem: statement

We now come to something much more surprising.

Recall that a multidigraph $D = (V, A, \varphi)$ 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, this is also sufficient (by Theorem 4.7.2 **(a)**).

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

Theorem 5.9.1 (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 of this theorem is an abbreviation for de Bruijn, van Aardenne–Ehrenfest, Smith and Tutte, who discovered it in the middle of the 20th century[45]. [46]

[45]More precisely, van Aardenne–Ehrenfest and de Bruijn discovered it in 1951 (see [VanEhr51, §6]) generalizing an earlier result of Smith and Tutte.

[46]We note that the number of Eulerian circuits of D whose last arc is a is precisely the number

To prove this theorem, we shall restate it in terms of "arborescences to" (as opposed to "arborescences from"). Mathematically speaking, this restatement isn't really necessary (the argument is the same in both cases up to reversing the directions of all arcs), but it helps make the proof more intuitive, since it lets us build our Eulerian circuits by moving forwards rather than backwards.

5.10. Arborescences rooted to r

Here is the formal definition of "arborescences to":

Definition 5.10.1. Let D be a multidigraph. Let r be a vertex of D.

(a) 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.

(b) 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, Definition 5.6.1 and Definition 5.10.1 differ only in the direction of the arcs. In other words, if we reverse each arc of our digraph (turning its source into its target and vice versa), then a from-root becomes a to-root, and an arborescence rooted from r becomes an arborescence rooted to r, and vice versa. Thus, every property that we have proved for arborescences rooted from r can be translated into the language of arborescences rooted to r by reversing all arcs.

If you want to see this stated more rigorously, here is a formal definition of "reversing each arc":

Definition 5.10.2. Let $D = (V, A, \psi)$ be a multidigraph. Then, D^{rev} shall denote the multidigraph $(V, A, \tau \circ \psi)$, where $\tau : V \times V \to V \times V$ is the map that sends each pair (s, t) to (t, s). Thus, if an arc a of D has source s and target t, then it is also an arc of D^{rev}, but in this digraph D^{rev} it has source t and target s.

The multidigraph D^{rev} is called the **reversal** of the multidigraph D; we say that it is obtained from D by "reversing each arc".

This notion of "reversing each arc" allows us to reverse walks in digraphs: If $\mathbf{w}$ is a walk from a vertex s to t in some multidigraph D, then its reversal $\operatorname{rev} \mathbf{w}$ (obtained by reading $\mathbf{w}$ backwards) is a walk from t to s in the multidigraph D^{rev}. The same holds if we replace the word "walk" by "path". Thus, we easily obtain the following:

Proposition 5.10.3. Let D be a multidigraph. Let r be a vertex of D. Then:

(a) The vertex r is a to-root of D if and only if r is a from-root of D^{rev}.

---

of all Eulerian circuits of D counted up to rotation. Indeed, each Eulerian circuit of D contains the arc a exactly once, and thus can be rotated in a unique way to end with a.

(b) The digraph D is an arborescence rooted to r if and only if D^{rev} is an arborescence rooted from r.

Proof. Completely straightforward unpacking of the definitions. □

Note that when we reverse each arc in a digraph D, the outdegrees of its vertices become their indegrees and vice versa. Hence, a balanced digraph D remains balanced when this happens. In particular, the BEST theorem (Theorem 5.9.1) thus gets translated as follows:

Theorem 5.10.4 (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,

$$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^{+} u - 1\right)!.$$

We will soon prove Theorem 5.10.4, and then derive Theorem 5.9.1 from it by reversing the arcs.

First, however, let us state the analogue of the Arborescence Equivalence Theorem (Theorem 5.6.5) for "arborescences rooted to r" (as opposed to "arborescences rooted from r"):

Theorem 5.10.5 (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 multigraph 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 the vertex r will no longer be a to-root of the resulting multidigraph.
- **Statement A′6:** We have $\deg^{+} r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^{+} v = 1$.

Proof. Upon reversing all arcs of D, this turns into the original Arborescence Equivalence Theorem (Theorem 5.6.5). □

5.11. The BEST theorem: proof

We now come to the proof of the BEST theorem (Theorem 5.9.1). As we said, we proceed by proving Theorem 5.10.4 first. We first outline the idea of the proof; then we will give the details.

Proof idea for Theorem 5.10.4. An a**-Eulerian circuit** shall mean an Eulerian circuit of D whose first arc is a.

Let $\mathbf{e}$ be an a-Eulerian circuit. Its first arc is a; therefore, its first and last vertex is r.

Being an Eulerian circuit, $\mathbf{e}$ must contain each arc of D and therefore contain each vertex of D (since each vertex has outdegree > 0). For each vertex $u \neq r$, we let $e(u)$ be the **last exit** of $\mathbf{e}$ from u, that is, the last arc of $\mathbf{e}$ that has source u. Let $\operatorname{Exit}\mathbf{e}$ be the set of these last exits $e(u)$ for all vertices $u \neq r$. Then, we claim:

> *Claim 1:* This set $\operatorname{Exit}\mathbf{e}$ (or, more precisely, the spanning subdigraph $(V,\ \operatorname{Exit}\mathbf{e},\ \psi\mid_{\operatorname{Exit}\mathbf{e}})$) is a spanning arborescence of D rooted to r.

Let's assume for the moment that Claim 1 is proven. Thus, given any a-Eulerian circuit $\mathbf{e}$, we have constructed a spanning arborescence of D rooted to r.

How many a-Eulerian circuits $\mathbf{e}$ lead to a given arborescence in this way? The answer is rather nice:

> *Claim 2:* For each spanning arborescence $(V, B, \psi\mid_B)$ of D rooted to r, there are exactly $\prod\limits_{u\in V}\left(\deg^+ u - 1\right)!$ many a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit}\mathbf{e} = B$.

Let us again assume that this is proven. Combining Claim 1 with Claim 2, we obtain a $\prod\limits_{u\in V}\left(\deg^+ u - 1\right)!$-to-1 correspondence between the a-Eulerian circuits and the spanning arborescences of D rooted to r. Thus, the number of the former is $\prod\limits_{u\in V}\left(\deg^+ u - 1\right)!$ times the number of the latter. But this is precisely the claim of Theorem 5.10.4. Hence, in order to prove Theorem 5.10.4, it remains to prove Claim 1 and Claim 2. □

Here is the complete proof:

Proof of Theorem 5.10.4. Some notations first:

An **outgoing arc** from a vertex u will mean an arc whose source is u. An **incoming arc** into a vertex u will mean an arc whose target is u.

An a**-Eulerian circuit** shall mean an Eulerian circuit of D whose first arc is a.

A **sparb** shall mean a spanning arborescence of D rooted to r.

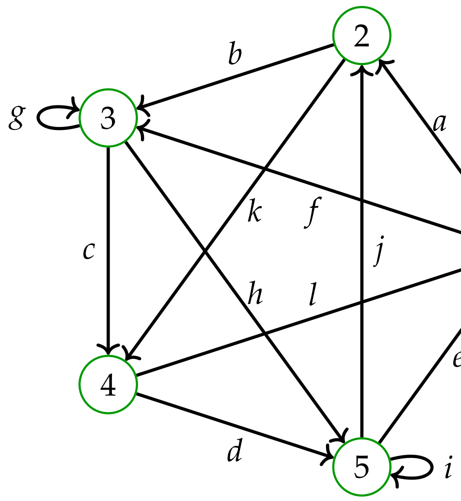
A spanning subdigraph of D always has the form $(V, B, \psi\mid_B)$ for some subset B of A. Thus, it is uniquely determined by its arc set B.

Hence, from now on, we shall identify a spanning subdigraph $(V, B, \psi\mid_B)$ of D with its arc set B. Conversely, any subset B of A will be identified with the corresponding spanning subdigraph $(V, B, \psi\mid_B)$ of D. Thus, for instance, when we say that a subset B of A "is a sparb", we shall actually mean that the corresponding spanning subdigraph $(V, B, \psi\mid_B)$ is a sparb.

For each a-Eulerian circuit $\mathbf{e}$, we define a subset $\operatorname{Exit}\mathbf{e}$ of A as follows:

Let $\mathbf{e}$ be an a-Eulerian circuit. Its first arc is a; thus, its first and last vertex is r. Being an Eulerian circuit, $\mathbf{e}$ must contain each arc of D and therefore also contain each vertex of D (since each vertex of D has outdegree > 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 $u \in V \setminus \{r\}$. Thus, we have defined a subset $\operatorname{Exit}\mathbf{e}$ of A for each a-Eulerian circuit $\mathbf{e}$.

Example 5.11.1. Here is an example of this construction: Let D be the multidigraph



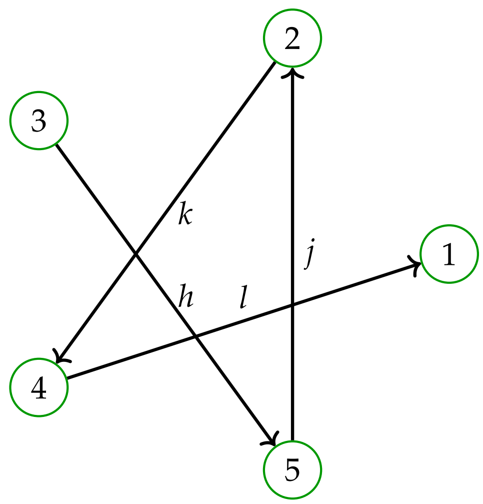
with $r = 1$, and let $\mathbf{e}$ be the a-Eulerian circuit

$$(1, a, 2, b, 3, c, 4, d, 5, e, 1, f, 3, g, 3, h, 5, i, 5, j, 2, k, 4, l, 1)$$

(we have deliberately named the arcs in such a way that they appear on an Eulerian circuit in alphabetic order). Then,

$$e(2) = k, \qquad e(3) = h, \qquad e(4) = l, \qquad e(5) = j,$$

so that $\operatorname{Exit}\mathbf{e} = \{k, h, l, j\}$. Here is $\operatorname{Exit}\mathbf{e}$ as a spanning subdigraph:



Now, we claim the following:

Claim 1: Let $\mathbf{e}$ be an a-Eulerian circuit. Then, the set $\operatorname{Exit}\mathbf{e}$ is a sparb.

Claim 2: For each sparb B (regarded as a subset of A), there are exactly $\prod\limits_{u \in V} (\deg^+ u - 1)!$ many a-Eulerian circuits $\mathbf{e}$ such that $\operatorname{Exit}\mathbf{e} = B$.

[*Proof of Claim 1:* The set $\operatorname{Exit}\mathbf{e}$ contains exactly one outgoing arc (namely, $e(u)$) from each vertex $u \in V \setminus \{r\}$, and no outgoing arc from r. Thus, $|\operatorname{Exit}\mathbf{e}| = |V| - 1$.

Let us number the arcs of $\mathbf{e}$ as $a_1, a_2, \ldots, a_m$, in the order in which they appear in $\mathbf{e}$. (Thus, $a_1 = a$, since the first arc of $\mathbf{e}$ is a.)

Recall that the arcs in $\operatorname{Exit}\mathbf{e}$ are the arcs $e(u)$ for all $u \in V \setminus \{r\}$ (defined as above – i.e., the arc $e(u)$ is the last exit of $\mathbf{e}$ from u). We shall refer to these arcs as the **last-exit arcs**.

For each $u \in V \setminus \{r\}$, we let $j(u)$ be the unique number $i \in \{1, 2, \ldots, m\}$ such that $e(u) = a_i$. (This i indeed exists and is unique, since each arc of D appears exactly once on $\mathbf{e}$.) Thus, $j(u)$ tells us how late in the Eulerian circuit $\mathbf{e}$ the arc $e(u)$ appears. Since $e(u)$ is the last exit of $\mathbf{e}$ from u, the Eulerian circuit $\mathbf{e}$ never visits the vertex u again after this.

Thus, if a last-exit arc $e(u)$ has target $v \neq r$, then

$$j(u) < j(v) \tag{21}$$

(because the arc $e(u)$ leads the circuit $\mathbf{e}$ into the vertex v, which the circuit then has to exit at least once; therefore, the corresponding last-exit arc $e(v)$ has to appear later in $\mathbf{e}$ than the arc $e(u)$).

We shall now show that r is a to-root of Exit $\mathbf{e}$ (that is, of the spanning subdigraph $(V, \operatorname{Exit} \mathbf{e}, \psi\mid_{\operatorname{Exit}\mathbf{e}})$). To this purpose, we must show that for each vertex $v \in V$, there is a path from v to r in the digraph $(V, \operatorname{Exit} \mathbf{e}, \psi\mid_{\operatorname{Exit}\mathbf{e}})$.

Indeed, let $v \in V$ be any vertex. We must find a path from v to r in the digraph $(V, \operatorname{Exit} \mathbf{e}, \psi\mid_{\operatorname{Exit}\mathbf{e}})$. It will suffice to find a walk from v to r in this digraph (by Corollary 4.5.8). In other words, we must find a way to walk from v to r in D using last-exit arcs only.

So we start walking at v. If $v = r$, then we are already done. Otherwise, we have $v \in V \setminus \{r\}$, so that the arc $e(v)$ and the number $j(v)$ are well-defined. We thus take the arc $e(v)$. This brings us to a vertex v' (namely, the target of $e(v)$) that satisfies $j(v) < j(v')$ (by (21)). If this vertex v' is r, then we are done. If not, then $e(v')$ and $j(v')$ are well-defined, so we continue our walk by taking the arc $e(v')$. This brings us to a further vertex v'' (namely, the target of $e(v')$) that satisfies $j(v') < j(v'')$ (by (21)). If this vertex v'' is r, then we are done. Otherwise, we proceed as before. We thus construct a walk

$$(v, e(v), v', e(v'), v'', e(v''), \ldots)$$

that either goes on indefinitely or stops at the vertex r.

However, this walking process cannot go on forever (since the chain of inequalities $j(v) < j(v') < j(v'') < \cdots$ would force the numbers $j(v), j(v'), j(v''), \ldots$ to be all distinct, but there are only m distinct numbers in $\{1, 2, \ldots, m\}$). Thus, it must stop at the vertex r. So we have found a walk from v to r using last-exit arcs only. Thus, Exit $\mathbf{e}$ has a walk from v to r. Hence, Exit $\mathbf{e}$ has a path from v to r.

Forget that we fixed v. We thus have shown that for each vertex $v \in V$, there is a path from v to r in the digraph $(V, \operatorname{Exit} \mathbf{e}, \psi\mid_{\operatorname{Exit}\mathbf{e}})$. In other words, r is a to-root of Exit $\mathbf{e}$. Hence, we conclude (using the implication A'2$\Longrightarrow$A'1 in Theorem 5.10.5) that Exit $\mathbf{e}$ is an arborescence rooted to r (since $|\operatorname{Exit} \mathbf{e}| = |V| - 1$). Therefore, Exit $\mathbf{e}$ is a sparb. This proves Claim 1.]

[*Proof of Claim 2:* Let B be a sparb. (As before, B is a set of arcs, and we identify it with the spanning subdigraph $(V, B, \psi\mid_B)$.)

We must prove that there are exactly $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$ many a-Eulerian circuits $\mathbf{e}$ such that Exit $\mathbf{e} = B$.

We shall refer to the arcs in B as the **B-arcs**. Recall that B is an arborescence rooted to r (since B is a sparb). Hence, by the implication A'1$\Longrightarrow$A'6 in Theorem 5.10.5, we see that the outdegrees of its vertices satisfy

$$\deg^+_B r = 0, \qquad \text{and} \qquad \deg^+_B v = 1 \text{ for all } v \in V \setminus \{r\}$$

(where $\deg^+_B v$ means the outdegree of a vertex in the digraph $(V, B, \psi\mid_B)$). In other words, 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 Exit $\mathbf{e} = B$.

Let us try to construct such an a-Eulerian circuit **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 would walk on a digraph: Each time it reaches a vertex, it chooses an arbitrary arc leading out of this vertex, observing the following two rules:

1. It never uses an arc that it has already used before.

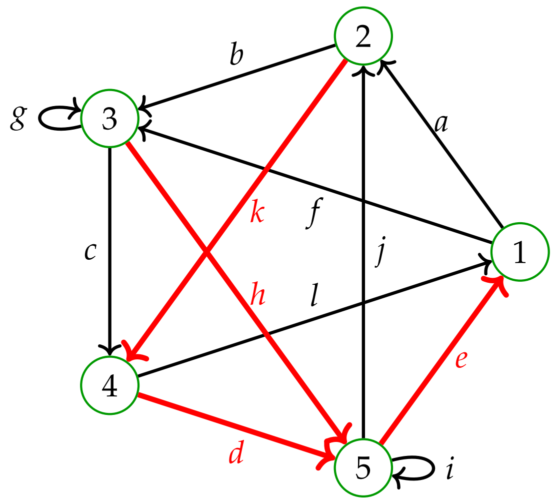
2. It never uses a B-arc unless it has to (i.e., unless this B-arc is the only outgoing arc from its current position that is still unused).

Clearly, the turtle will eventually get stuck at some vertex (with no more arcs left to continue walking along), since D has only finitely many arcs.

Let **w** be the total walk that the turtle has traced by the time it got stuck. Thus, **w** is a trail (i.e., a walk that uses no arc more than once) that starts with the vertex r and the arc a.

We will soon see that **w** is an a-Eulerian circuit satisfying $\operatorname{Exit}\mathbf{w} = B$. First, however, let us see an example:

Example 5.11.2. Let D be the multidigraph



,

and let $r = 1$ and $a = a$ (we called it a on purpose). Let B be the set $\{d, e, h, k\}$, regarded as a spanning subdigraph of D. (The arcs of B are drawn bold and in red in the above picture.)

The turtle starts at $r = 1$ and walks along the arc a. This leads it to the vertex 2. It now must choose between the arcs b and k, but since it must not use the B-arc k unless it has to, it is actually forced to take the arc b next. This brings it to the vertex 3. It now has to choose between the arcs c, g and h, but again the arc h is disallowed because it is not yet time to use a B-arc. Let us say that it takes the arc g. This brings it back to the vertex 3. Next, the turtle must walk along c (since g is already used, while the B-arc still must wait until it is the only option). This brings it to the vertex 4. Its next step is to take the arc l to the vertex 1. From there, it follows the arc f to the vertex 3. Now, it can finally take the B-arc h, since all the other outgoing arcs from 3 have already been used. This brings it to the vertex 5. Now it has a choice between the arcs e, i and j, but the arc e is disallowed because it is a B-arc. Let us say it decides to use the arc j. This brings it to the vertex 2. From there, it takes the B-arc k to the vertex 4 (since it has no other options). From there, it continues along the B-arc d to the vertex 5. Now, it has to traverse the loop i, and then leave 5 along the B-arc e to come back to 1. At this point, the turtle is stuck, since it has nowhere left to go. The walk $\mathbf{w}$ we obtained is thus

$$\mathbf{w} = (1, a, 2, b, 3, g, 3, c, 4, l, 1, f, 3, h, 5, j, 2, k, 4, d, 5, i, 5, e, 1).$$

(Of course, other choices would have led to other walks.)

Returning to the general case, let us analyze the walk $\mathbf{w}$ traversed by the turtle.

- First, we claim that $\mathbf{w}$ is a closed walk (i.e., ends at r).

  [*Proof:* Assume the contrary. Let u be the ending point of $\mathbf{w}$. Thus, u is the vertex at which the turtle gets stuck. Moreover, $u \neq r$ (since we just assumed that $\mathbf{w}$ is not a closed walk). Hence, the walk $\mathbf{w}$ enters the vertex u more often than it leaves it (since it ends but does not start at u). In other words, the turtle has entered the vertex u more often than it has left it. However, since D is balanced, we have $\deg^- u = \deg^+ u$. The turtle has entered the vertex u at most $\deg^- u$ times (because it cannot use an arc twice, but there are only $\deg^- u$ many arcs with target u). Thus, it has left the vertex u **less** than $\deg^- u$ times (because it has entered the vertex u more often than it has left it). Since $\deg^- u = \deg^+ u$, this means that the turtle has left the vertex u less than $\deg^+ u$ times. Thus, by the time the turtle has gotten stuck at u, there is at least one outgoing arc from u that has not been used by the turtle. Therefore, the turtle is not actually stuck at u. This is a contradiction. Thus, our assumption was wrong, so we have proved that $\mathbf{w}$ is a closed walk.]

In other words, $\mathbf{w}$ is a circuit. We shall next show that $\mathbf{w}$ is an Eulerian circuit.

To do so, we introduce one more piece of notation: A vertex u of D will be called **exhausted** if the turtle has used each outgoing arc from u (that is, if each outgoing arc from u is used in the circuit $\mathbf{w}$).

Since $\mathbf{w}$ is a circuit, the ending point of $\mathbf{w}$ is its starting point, i.e., the vertex r. Thus, the turtle must have gotten stuck at r. Hence, the vertex r is exhausted.

- We shall now show that **all** vertices of D are exhausted.

[*Proof:* Assume the contrary. Thus, there exists a vertex u of D that is not exhausted. Consider this u. But B is a sparb, thus an arborescence rooted to r. Hence, r is a to-root of B. Therefore, there exists a path $\mathbf{p} = (p_0, b_1, p_1, b_2, p_2, \ldots, b_k, p_k)$ from u to r in B. Consider this path. Thus, we have $p_0 = u$ and $p_k = r$, and all the arcs $b_1, b_2, \ldots, b_k$ belong to B.

There exists at least one $i \in \{0, 1, \ldots, k\}$ such that the vertex p_i is exhausted (for instance, $i = k$ qualifies, since $p_k = r$ is exhausted). Consider the **smallest** such i. Then, $p_i \neq p_0$ (since p_i is exhausted, but $p_0 = u$ is not). Hence, $i \neq 0$, so that $i \geq 1$. Therefore, p_{i-1} exists. Moreover, the vertex p_{i-1} is not exhausted (since i was defined to be the **smallest** element of $\{0, 1, \ldots, k\}$ such that p_i is exhausted).

The arc b_i has source p_{i-1} and target p_i. Thus, it is an outgoing arc from p_{i-1} and incoming arc into p_i. Furthermore, it belongs to B (since all the arcs $b_1, b_2, \ldots, b_k$ belong to B).

The digraph D is balanced; thus, $\deg^+(p_i) = \deg^-(p_i)$.

The vertex p_i is exhausted. In other words, the turtle has used each outgoing arc from p_i (by the definition of "exhausted"). Since the turtle never reuses an arc, this entails that the turtle has used exactly $\deg^+(p_i)$ many outgoing arcs from p_i (since $\deg^+(p_i)$ is the total number of outgoing arcs from p_i in D). In other words, it has used exactly $\deg^-(p_i)$ many outgoing arcs from p_i (since $\deg^+(p_i) = \deg^-(p_i)$).

However, the turtle's trajectory is a closed walk (in fact, it is the walk $\mathbf{w}$, which is closed). Thus, it must enter the vertex p_i as often as it leaves this vertex. In other words, the number of incoming arcs into p_i used by the turtle must equal the number of outgoing arcs from p_i used by the turtle. Since we just found (in the preceding paragraph) that the latter number is $\deg^-(p_i)$, we thus conclude that the former number is $\deg^-(p_i)$ as well. In other words, the turtle must have used exactly $\deg^-(p_i)$ many incoming arcs into p_i. Since $\deg^-(p_i)$ is the total number of incoming arcs into p_i in D, we thus conclude that the turtle must have used all incoming arcs into p_i (since the turtle never reuses an arc).

Hence, in particular, the turtle must have used the arc b_i (since b_i is an incoming arc into p_i). This arc b_i is an outgoing arc from p_{i-1}. But b_i is a B-arc, and thus our turtle uses this arc only as a last resort (i.e., after using all other outgoing arcs from p_{i-1}). Hence, we conclude that the turtle must have used all outgoing arcs from p_{i-1} (since it has used b_i). In other words, p_{i-1} is exhausted. But this contradicts the fact that p_{i-1} is not exhausted! This shows that our assumption was wrong, and our proof is finished.[47]].]

[47]For the sake of diversity, let me sketch a *second proof* of the same claim (i.e., that all vertices in D are exhausted):

Assume the contrary. Thus, there exists a non-exhausted vertex u of D. Consider this u.

Thus, we have shown that all vertices of D are exhausted. In other words, the turtle has used all arcs of D. In other words, the trail $\mathbf{w}$ contains all arcs of D. Since $\mathbf{w}$ is a trail and a closed walk, this entails that $\mathbf{w}$ is an Eulerian circuit of D. Since $\mathbf{w}$ starts with r and a, this shows further that $\mathbf{w}$ is an a-Eulerian circuit. Since the turtle only used B-arcs as a last resort (and it used each B-arc eventually, because $\mathbf{w}$ is Eulerian), we have $\operatorname{Exit} \mathbf{w} = B$.

Thus, the turtle's walk has produced an a-Eulerian circuit $\mathbf{e}$ satisfying $\operatorname{Exit} \mathbf{e} = B$ (namely, the walk $\mathbf{w}$). However, this circuit depends on some decisions the turtle made during its walk. Namely, every time the turtle was at some vertex $u \in V$, it had to decide which arc to take next; this arc had to be an unused arc with source u, subject to the conditions that

1. if $u \neq r$, then the B-arc[48] has to be used last;

2. if $u = r$, then the arc a has to be used first.

Let us count how many options the turtle has had in total. To make the argument clearer, we modify the procedure somewhat: Instead of deciding ad-hoc which arc to take, the turtle should now make all these decisions before embarking on its journey. To do so, it chooses, for each vertex $u \in V$, a total order on the set of all arcs with source u, such that

1. if $u \neq r$, then the B-arc comes last in this order, and

Then, $u \neq r$ (since r is exhausted but u is not). Since u is not exhausted, there is at least one outgoing arc from u that the turtle has not used. Hence, the turtle has not used the B-arc outgoing from u (since the turtle never uses a B-arc before it has to). Let f be this B-arc, and let u' be its target. Thus, the turtle has not used all incoming arcs of u' (because it has not used the arc f). As a consequence, it has not used all outgoing arcs from u' either (because the turtle has left u' as often as it has entered u', but the balancedness of D entails that $\deg^{-}(u') = \deg^{+}(u')$). In other words, the vertex u' is non-exhausted.

Thus, by starting at the non-exhausted vertex u and taking the B-arc outgoing from u, we have arrived at a further non-exhausted vertex u'. Applying the same argument to u' instead of u, we can take a further B-arc and arrive at a further non-exhausted vertex u''. Continuing like this, we obtain an infinite sequence $(u, u', u'', \ldots)$ of non-exhausted vertices such that any vertex in this sequence is reached from the previous one by traveling along a B-arc. Clearly, this sequence must have two equal vertices (since D has only finitely many vertices). For example, let's say that $u'' = u'''''$. Then, if we consider only the part of the sequence between u'' and u''''', then we obtain a closed walk

$$(u'', *, u''', *, u'''', *, u''''') ,$$

where each asterisk stands for some B-arc (not the same one, of course). This is a closed walk of the digraph $(V, B, \psi\mid_B)$. Since this closed walk has length > 0, it cannot be a path; therefore, it contains a cycle (by Proposition 4.5.9). Thus, we have found a cycle of the digraph $(V, B, \psi\mid_B)$. However, the digraph $(V, B, \psi\mid_B)$ is an arborescence, and thus has no cycles (because if D is an arborescence, then any cycle of D would be a cycle of D^{und}; but the multigraph D^{und} has no cycles by the definition of an arborescence). The previous two sentences contradict each other. This shows that our assumption was wrong, and our proof is finished.

[48] We say "**the** B-arc", because there is exactly one B-arc with source u.

2. if $u = r$, then the arc a comes first in this order.

Note that this total order can be chosen in $\left(\deg^+ u - 1\right)!$ many ways (since there are $\deg^+ u$ arcs with source u, and we can freely choose their order except that one of them has a fixed position). Thus, in total, there are $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$ many options for how the turtle can choose all these orders. Once these orders have been chosen, the turtle then uses them to decide which arcs to walk along: Namely, the first time it visits the vertex u, it leaves it along the first arc (according to its chosen order); the second time, it uses the second arc; the third time, the third arc; and so on.

So the turtle has $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$ many options, and each of these options leads to a different a-Eulerian circuit $\mathbf{e}$ (because the total orders chosen by the turtle are reflected in $\mathbf{e}$: they are precisely the orders in which the respective arcs appear in $\mathbf{e}$). Moreover, each a-Eulerian circuit $\mathbf{e}$ satisfying $\operatorname{Exit} \mathbf{e} = B$ comes from one of these options[49].

Therefore, the total number of a-Eulerian circuits $\mathbf{e}$ satisfying $\operatorname{Exit} \mathbf{e} = B$ is the total number of options, which is $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$ as we know. This proves Claim 2.]

With Claims 1 and 2 proved, we are almost done. The map

$$\{a\text{-Eulerian circuits of } D\} \to \{\text{sparbs}\}, \qquad \mathbf{e} \mapsto \operatorname{Exit} \mathbf{e}$$

is well-defined (by Claim 1). Furthermore, Claim 2 shows that this map is a $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$-to-1 correspondence[50] (i.e., each sparb B has exactly $\prod\limits_{u \in V} \left(\deg^+ u - 1\right)!$ many preimages under this map). Thus, by the multijection principle[51], we conclude that[52]

$$(\text{\# of } a\text{-Eulerian circuits of } D) = \left(\prod_{u \in V} \left(\deg^+ u - 1\right)!\right) \cdot (\text{\# of sparbs}).$$

[49] *Proof.* Let $\mathbf{e}$ be an a-Eulerian circuit satisfying $\operatorname{Exit} \mathbf{e} = B$. Then, by choosing the appropriate total orders ahead of its journey, the turtle will trace this exact circuit $\mathbf{e}$. (Of course, the "appropriate total orders" are the ones dictated by $\mathbf{e}$: That is, for each vertex $u \in V$, the turtle must pick the same total order on the set of all arcs with source u in which they appear on $\mathbf{e}$. This choice is legitimate, because the arc a is the first arc of $\mathbf{e}$ (so it will certainly come first in its order), and because each B-arc appears in $\mathbf{e}$ after all other arcs from the same source have appeared (so it will come last in its total order).)

[50] An m**-to-1 correspondence** (where m is a nonnegative integer) means a map $f : X \to Y$ between two sets such that each element of Y has exactly m preimages under f.

[51] The **multijection principle** is a basic counting principle that says the following: Let X and Y be two finite sets, and let $m \in \mathbb{N}$. Let $f : X \to Y$ be an m-to-1 correspondence (i.e., a map such that each element of Y has exactly m preimages under f). Then, $|X| = m \cdot |Y|$.

For example, n (intact) sheep have $4n$ legs in total, since the map that sends each leg to its sheep is a 4-to-1 correspondence.

[52] The symbol "#" means "number".

Since $\varepsilon(D,a) = (\#\text{ of } a\text{-Eulerian circuits of } D)$ and $\tau(D,r) = (\#\text{ of sparbs})$, we can rewrite this as follows:

$$\varepsilon(D,a) = \left(\prod_{u\in V}\left(\deg^+ u - 1\right)!\right)\cdot\tau(D,r) = \tau(D,r)\cdot\prod_{u\in V}\left(\deg^+ u - 1\right)!.$$

This proves Theorem 5.10.4. □

Proof of Theorem 5.9.1. As we already mentioned, Theorem 5.9.1 follows from Theorem 5.10.4 by reversing each arc (i.e., by applying Theorem 5.10.4 to the digraph D^{rev} instead of D). □

5.12. A corollary about spanning arborescences

Before we actually use the BEST (or BEST') theorem to count the Eulerian circuits on any digraph, let us mention a neat corollary for the number of spanning arborescences:

Corollary 5.12.1. 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 of Corollary 5.12.1. WLOG assume that $|V| > 1$ (else, the claim is obvious). If there is a vertex $v \in V$ with $\deg^+ v = 0$, then this vertex v satisfies $\deg^- v = 0$ as well (since the balancedness of D entails $\deg^- v = \deg^+ v = 0$), and therefore D has no spanning arborescences at all (since any spanning arborescence would have an arc with source or target v). Thus, we WLOG assume that $\deg^+ v > 0$ for all $v \in V$. In other words, each vertex has outdegree > 0.

Let r and s be two vertices of D. We must prove that $\tau(D,r) = \tau(D,s)$.

Pick an arc a with source r. (This exists, since $\deg^+ r > 0$.) Pick an arc b with source s. (This exists, since $\deg^+ s > 0$.)

Applying the BEST' theorem (Theorem 5.10.4), we get

$$\varepsilon(D,a) = \tau(D,r)\cdot\prod_{u\in V}\left(\deg^+ u - 1\right)! \qquad \text{and similarly}$$
$$\varepsilon(D,b) = \tau(D,s)\cdot\prod_{u\in V}\left(\deg^+ u - 1\right)!.$$

However, $\varepsilon(D,a) = \varepsilon(D,b)$, since counting Eulerian circuits that start with a is equivalent to counting Eulerian circuits that start with b (because an Eulerian circuit can be rotated uniquely to start with any given arc). Thus, we obtain

$$\tau(D,r)\cdot\prod_{u\in V}\left(\deg^+ u - 1\right)! = \varepsilon(D,a) = \varepsilon(D,b) = \tau(D,s)\cdot\prod_{u\in V}\left(\deg^+ u - 1\right)!.$$

Cancelling the (nonzero!) number $\prod_{u\in V}\left(\deg^+ u - 1\right)!$ from this equality, we obtain $\tau(D,r) = \tau(D,s)$. This proves Corollary 5.12.1. □

5.13. Spanning arborescences vs. spanning trees

The BEST theorem (Theorem 5.10.4 or Theorem 5.9.1) connects the # of Eulerian circuits in a digraph with the # of spanning arborescences of the same digraph. Now let us try to find a way to compute the latter.

For example, let us try to do this for digraphs of the form G^{bidir} where G is a multigraph. I claim that the spanning arborescences of G^{bidir} rooted to a given vertex r are just the spanning trees of G in disguise:

Proposition 5.13.1. 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

$$\overline{B} := \{e \mid (e, i) \in B\}.$$

(Recall that we are identifying spanning arborescences with their arc sets, so that "$(e, i) \in B$" means "(e, i) is an arc of B".)

Then:

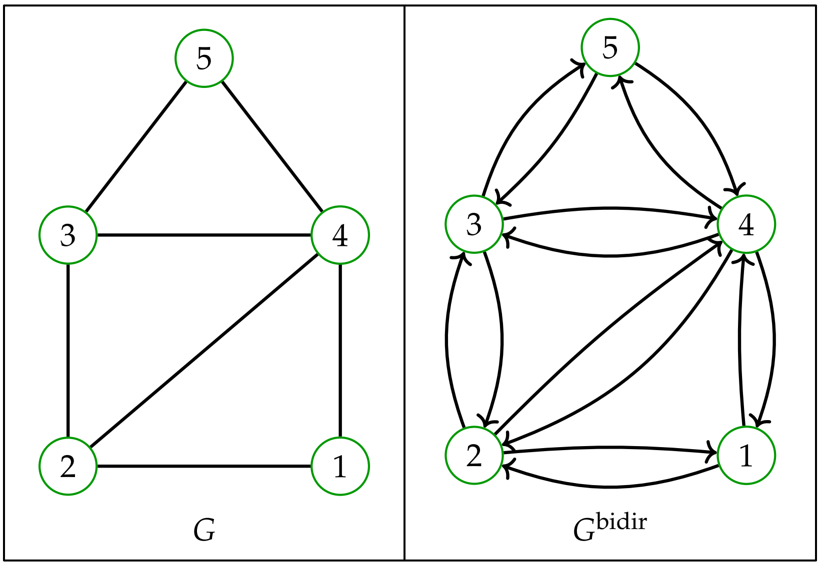
(a) If B is a spanning arborescence of G^{bidir} rooted to r, then $\overline{B}$ is a spanning tree of G.

(b) The map

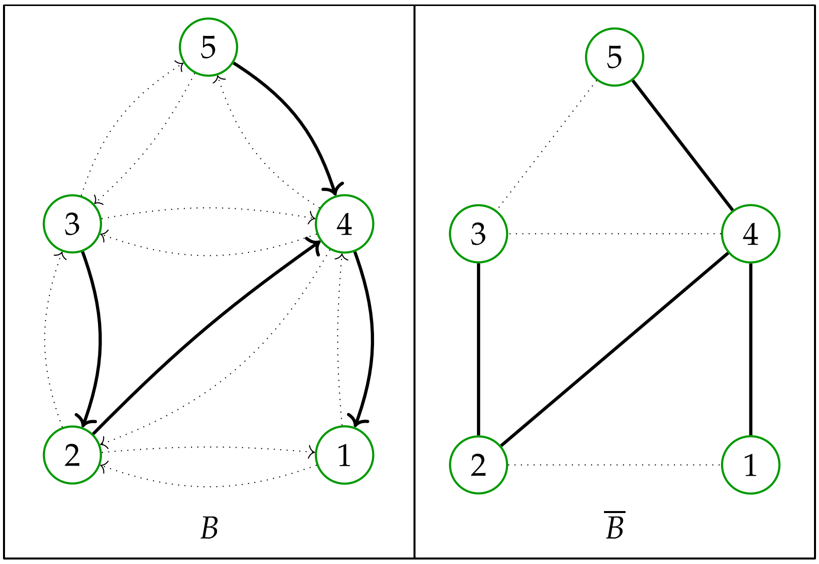
$$\left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} \to \{\text{spanning trees of } G\},$$
$$B \mapsto \overline{B}$$

is a bijection.

Example 5.13.2. Here is a multigraph G (on the left) with the corresponding multidigraph G^{bidir} (on the right):



Here is a spanning arborescence B of G^{bidir} rooted to 1, and the corresponding spanning tree $\overline{B}$ of G:



(here, the arcs of G^{bidir} that **don't** belong to B, as well as the edges of G that **don't** belong to $\overline{B}$, have been drawn as dotted arrows). It is fairly easy to see how B can be reconstructed from $\overline{B}$: You just need to replace each edge of $\overline{B}$ by the appropriately directed arc (namely, the one that is "directed towards 1").

Proof of Proposition 5.13.1. This is an exercise in yak-shaving (and we have, in fact, shaved a very similar yak in Section 5.7; the only difference is that we are no longer dealing with trees in isolation, but rather with spanning trees of G).

(a) Let B be a spanning arborescence of G^{bidir} rooted to r. Then, B^{und} is a tree (by the implication A'1$\Longrightarrow$A'3 in Theorem 5.10.5). However, it is easy to see that $B^{\text{und}} \cong \overline{B}$ as multigraphs (indeed, each vertex v of B^{und} corresponds to the same vertex v of $\overline{B}$, whereas any edge (e,i) of B^{und} corresponds to the edge e of $\overline{B}$) [53]. Thus, $\overline{B}$ is a tree (since B^{und} is a tree)[54], therefore a spanning tree of G (since $\overline{B}$ is clearly a spanning subgraph of G). This proves Proposition 5.13.1 **(a)**.

(b) We must prove that this map is surjective and injective.

Surjectivity: Let T be a spanning tree of G. Then, the multidigraph $T^{r\rightarrow}$ (defined in Definition 5.7.4) is an arborescence rooted from r (by Lemma 5.7.7). Reversing each arc in this arborescence $T^{r\rightarrow}$, we obtain a new multidigraph $T^{r\leftarrow}$, which is thus an arborescence rooted to r. Unfortunately, $T^{r\leftarrow}$ is not a subdigraph of G^{bidir}, for a rather stupid reason: The arcs of $T^{r\leftarrow}$ are elements of E, whereas the arcs of G^{bidir} are pairs of the form (e,i) with $e \in E$ and $i \in \{1,2\}$.

Fortunately, this is easily fixed: For each arc e of $T^{r\leftarrow}$, we let e' be the arc (e,i) of G^{bidir} that has the same source as e (and thus the same target as e). This is uniquely determined, since the arcs $(e,1)$ and $(e,2)$ of G^{bidir} have different sources[55]. If we replace each arc e of $T^{r\leftarrow}$ by the corresponding arc e' of G^{bidir}, then we obtain a spanning subdigraph S of G^{bidir} that is an arborescence rooted to r (since $T^{r\leftarrow}$ is an arborescence rooted to r, and we have only replaced its arcs by equivalent ones with the same sources and the same targets). In other words, we obtain a spanning arborescence S of G^{bidir} rooted to r. It is easy to see that $\overline{S} = T$. Hence, the map

$$\left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} \to \{\text{spanning trees of } G\},$$
$$B \mapsto \overline{B}$$

[53]Here we need to use the fact that for each edge e of $\overline{B}$, **exactly one** of the two pairs $(e,1)$ and $(e,2)$ is an edge of B^{und}. But this is easy to check: At least one of the two pairs $(e,1)$ and $(e,2)$ must be an arc of B (since e is an edge of $\overline{B}$). In other words, at least one of the two pairs $(e,1)$ and $(e,2)$ must be an edge of B^{und}. But both of these pairs cannot be edges of B^{und} at the same time (since this would create a cycle, but B^{und} is a tree and thus has no cycles). Hence, exactly one of these pairs is an edge of B^{und}, qed.

[54]Alternatively, you can prove this as follows: The vertex r is a to-root of B (since B is an arborescence rooted to r). Thus, for each $v \in V$, there is a path from v to r in B. By "projecting" this path onto $\overline{B}$ (that is, replacing each arc (e,i) of this path by the corresponding edge e of $\overline{B}$), we obtain a path from v to r in $\overline{B}$. This shows that the multigraph $\overline{B}$ is connected. Furthermore, the definition of $\overline{B}$ shows that $\left|\overline{B}\right| \leq |B| = |V| - 1$ (by Statement A'2 in Theorem 5.10.5, since B is an arborescence rooted to r). Hence, $\left|\overline{B}\right| < |V|$. Thus, we can apply the implication T5$\Longrightarrow$T1 of the Tree Equivalence Theorem (Theorem 5.2.4) to conclude that $\overline{B}$ is a tree.

[55]*Proof.* The edge e of T is not a loop (because T is a tree, but a tree cannot have any loops). Hence, its two endpoints are distinct. Thus, the arcs $(e,1)$ and $(e,2)$ of G^{bidir} have different sources (since their sources are the two endpoints of e).

sends S to T. This shows that T is a value of this map. Since we have proved this for every spanning tree T of G, we have thus shown that this map is surjective.

Injectivity: The main idea is that, in order to recover a spanning arborescence B back from the corresponding spanning tree $\overline{B}$, we just need to "orient the edges of the tree towards r". Here are the (annoyingly long) details:

Let B and C be two sparbs[56] such that $\overline{B} = \overline{C}$. We must show that $B = C$.

Assume the contrary. Thus, $B \neq C$. Let T be the tree $\overline{B} = \overline{C}$. Thus, each edge e of T corresponds to either an arc $(e,1)$ or an arc $(e,2)$ in B (since $T = \overline{B}$), and likewise for C. Conversely, each arc (e,i) of B or of C corresponds to an edge e of T. Hence, from $B \neq C$, we see that there must exist an edge e of T such that

- **either** we have $(e,1) \in B$ and $(e,2) \in C$,
- **or** we have $(e,1) \in C$ and $(e,2) \in B$.

Consider this edge e. We WLOG assume that $(e,1) \in B$ and $(e,2) \in C$ (else, we can just swap B with C). Let the arc $(e,1)$ of G^{bidir} have source s and target t, so that $(e,2)$ has source t and target s. The edge e thus has endpoints s and t.

Since B is an arborescence rooted to r, the vertex r is a to-root of B. Hence, there exists a path $\mathbf{p}$ from s to r in B. This path $\mathbf{p}$ must begin with the arc $(e,1)$ [57]. Projecting this path $\mathbf{p}$ down onto T, we obtain a path $\overline{\mathbf{p}}$ from s to r in T. (By the word "projecting", we mean replacing each arc (e,i) by the corresponding edge e. Clearly, doing this to a path in B yields a path in T, because $T = \overline{B}$.) Since the path $\mathbf{p}$ begins with the arc $(e,1)$, the "projected" path $\overline{\mathbf{p}}$ begins with the edge e. Thus, in the tree T, the path from s to r begins with the edge e (because this path must be the path $\overline{\mathbf{p}}$). As a consequence, t must be the second vertex of this path (since the edge e has endpoints s and t), so that removing the first edge from this path yields the path from t to r. Thus, $d(t,r) = d(s,r) - 1$, where d denotes distance on the tree T. Hence, $d(t,r) < d(s,r)$.

A similar argument (but with the roles of B and C swapped, as well as the roles of s and t swapped, and the roles of $(e,1)$ and $(e,2)$ swapped) shows that $d(s,r) < d(t,r)$. But this contradicts $d(t,r) < d(s,r)$.

This contradiction shows that our assumption was false. Thus, we have proved that $B = C$.

[56] Henceforth, "sparb" is short for "spanning arborescence of G^{bidir} rooted to r".

[57] *Proof.* Since r is a to-root of B, we know that there exists a path from t to r in B. Let $\mathbf{t}$ be this path. Extending this path $\mathbf{t}$ by the vertex s and the arc $(e,1)$ (which we both insert at the start of $\mathbf{t}$), we obtain a walk $\mathbf{t}'$ from s to r in B. (So, if $\mathbf{t} = (t, \ldots, r)$, then $\mathbf{t}' = (s, (e,1), t, \ldots, r)$.)

However, B is an arborescence rooted to r. Thus, Statement A'4 in the Dual Arborescence Equivalence Theorem (Theorem 5.10.5) shows that for each vertex $v \in V$, the digraph B has a unique walk from v to r. Hence, in particular, B has a unique walk from s to r. Thus, $\mathbf{p} = \mathbf{t}'$ (since both $\mathbf{p}$ and $\mathbf{t}'$ are walks from s to r in B). Since $\mathbf{t}'$ begins with the arc $(e,1)$, we thus conclude that $\mathbf{p}$ begins with the arc $(e,1)$.

Forget that we fixed B and C. We thus have shown that if B and C are two sparbs such that $\overline{B} = \overline{C}$, then $B = C$. In other words, our map

$$\begin{aligned} \left\{\text{spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right\} &\to \{\text{spanning trees of } G\}, \\ B &\mapsto \overline{B} \end{aligned}$$

is injective.

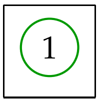
We have now shown that this map is both surjective and injective. Hence, it is a bijection. This proves Proposition 5.13.1 **(b)**. □

5.14. The matrix-tree theorem

5.14.1. Introduction

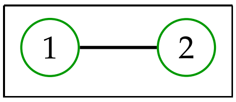
So counting spanning trees in a multigraph is a particular case of counting spanning arborescences (rooted to a given vertex) in a multidigraph. But how do we do either? Let us begin with some simple examples:

Example 5.14.1. There is only one spanning tree of the complete graph K_1:



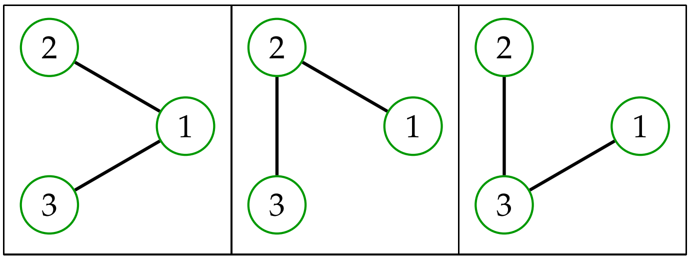
.

There is only one spanning tree of the complete graph K_2:



.

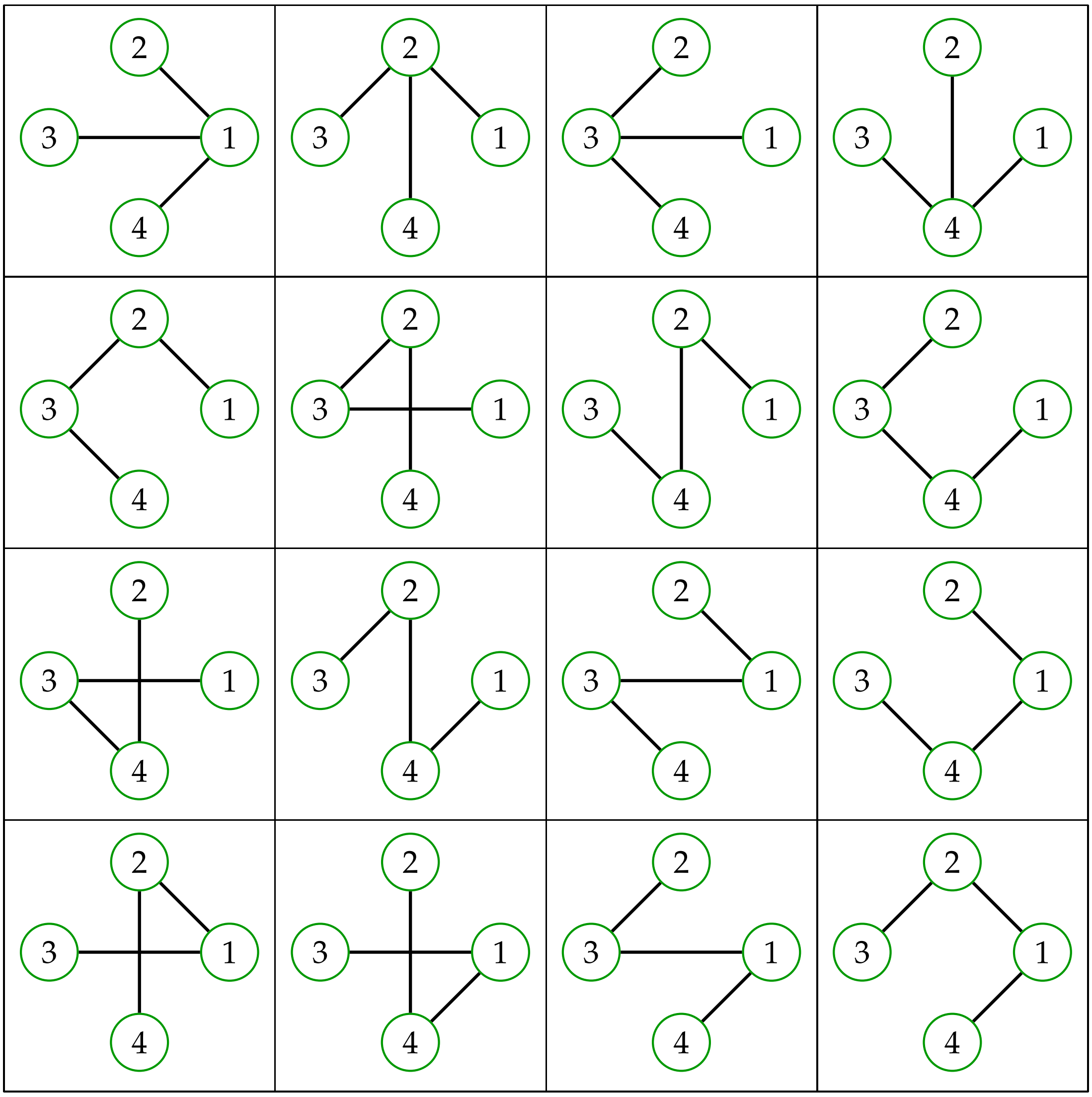
There are 3 spanning trees of the complete graph K_3:



.

(They are all isomorphic, but still distinct.)

There are 16 spanning trees of the complete graph K_4:



.

(There are only two non-isomorphic ones among them.)

This example suggests that the # of spanning trees of a complete graph K_n is n^{n-2}.

This is indeed true, and we will prove this later. For now, however, let us address the more general problem of counting spanning arborescences of an arbitrary digraph D.

5.14.2. Notations

First, we introduce a notation:

Definition 5.14.2. We will use the **Iverson bracket notation**: If $\mathcal{A}$ is any logical statement, then we set

$$[\mathcal{A}] := \begin{cases} 1, & \text{if } \mathcal{A} \text{ is true;} \\ 0, & \text{if } \mathcal{A} \text{ is false.} \end{cases}$$

For example, $[K_2 \text{ is a tree}] = 1$ whereas $[K_3 \text{ is a tree}] = 0$.

Definition 5.14.3. 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 with row i removed and column j removed.

For example,

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}_{2,3} = f \qquad \text{and} \qquad \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}_{\sim 2, \sim 3} = \begin{pmatrix} a & b \\ g & h \end{pmatrix}.$$

5.14.3. The Laplacian of a multidigraph

We shall now assign a matrix to (more or less) any multidigraph:[58]

Definition 5.14.4. 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 that have 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

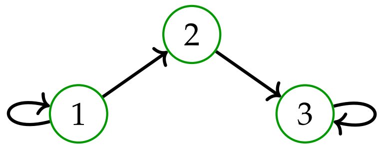
$$L_{i,j} = \left(\deg^+ i\right) \cdot \underbrace{[i = j]}_{\substack{\text{This is also} \\ \text{known as } \delta_{i,j}}} - a_{i,j} \qquad \text{for all } i, j \in V.$$

In other words, it is the matrix

$$L = \begin{pmatrix} \deg^+ 1 - a_{1,1} & -a_{1,2} & \cdots & -a_{1,n} \\ -a_{2,1} & \deg^+ 2 - a_{2,2} & \cdots & -a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} & \cdots & \deg^+ n - a_{n,n} \end{pmatrix}.$$

[58] Recall that the symbol "#" means "number".

Example 5.14.5. Let D be the digraph



.

Then, its Laplacian is

$$\begin{pmatrix} 2-1 & -1 & -0 \\ -0 & 1-0 & -1 \\ -0 & -0 & 1-1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix}.$$

One thing we notice from this example is that loops do not matter at all to the Laplacian L. Indeed, a loop with source i and target i counts once in $\deg^+ i$ and once in $a_{i,i}$, but these contributions cancel out.

Here is a simple property of Laplacians:

Proposition 5.14.6. 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

$$\begin{aligned} \sum_{j=1}^{n} L_{i,j} &= \sum_{j=1}^{n} \left(\left(\deg^+ i \right) \cdot [i = j] - a_{i,j} \right) \qquad \text{(by the definition of } L\text{)} \\ &= \underbrace{\sum_{j=1}^{n} \left(\deg^+ i \right) \cdot [i = j]}_{\substack{= \deg^+ i \\ \text{(since only the addend} \\ \text{for } j=i \text{ can be nonzero)}}} - \underbrace{\sum_{j=1}^{n} a_{i,j}}_{\substack{= \deg^+ i \\ \text{(since this is counting} \\ \text{all arcs with source } i)}} \\ &= \deg^+ i - \deg^+ i = 0. \end{aligned}$$

In other words, we have $Le = 0$ for the vector $e := (1, 1, \ldots, 1)^T$. Thus, this vector e lies in the kernel (aka nullspace) of L, and so L is singular.

(Note that we used the positivity of n here! If $n = 0$, then e is the zero vector, because a vector with 0 entries is automatically the zero vector.) □

5.14.4. The Matrix-Tree Theorem: statement

Proposition 5.14.6 shows that the determinant of the Laplacian of a digraph is not very interesting. It is common, however, that when a matrix has determinant 0, its largest nonzero minors (= determinants of submatrices) often carry some interesting information; they are "the closest the matrix has" to a nonzero determinant. In the case of the Laplacian, they turn out to count spanning arborescences:

Theorem 5.14.7 (Matrix-Tree Theorem)**.** 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 a vertex of D. Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det\left(L_{\sim r, \sim r}\right).$$

Before we prove this, some remarks:

- The determinant $\det\left(L_{\sim r, \sim r}\right)$ is the (r, r)-th entry of the adjugate matrix of L.
- 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 as $1, 2, \ldots, n$, and then this assumption will be satisfied. Thus, Theorem 5.14.7 helps you count the spanning arborescences of any digraph. That said, you can also drop the $V = \{1, 2, \ldots, n\}$ assumption from Theorem 5.14.7 if you are okay with matrices whose rows and columns are indexed not by numbers but by elements of an arbitrary finite set[59].

5.14.5. Application: Counting the spanning trees of K_n

Now, let us use the Matrix-Tree Theorem to count the spanning trees of K_n. This should provide some intuition for the theorem before we come to its proof.

We fix a positive integer n. Let L be the Laplacian of the multidigraph K_n^{bidir} (where K_n, as we recall, is the complete graph on the set $\{1, 2, \ldots, n\}$). Then, each vertex of K_n^{bidir} has outdegree $n-1$, and thus we have

$$L = \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}$$

(this is the $n \times n$-matrix whose diagonal entries are $n-1$ and whose off-diagonal entries are -1). By Proposition 5.13.1 **(b)** (applied to $G = K_n$ and $r = 1$), there is a bijection between $\left\{\text{spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right\}$ and

[59] Such matrices are perfectly fine, just somewhat unusual and hard to write down (which row do you put on top?). See `https://mathoverflow.net/questions/317105` for details.

{spanning trees of K_n}. Hence, by the bijection principle, we have

$$\begin{aligned}
&(\text{\# of spanning trees of } K_n) \\
&= \left(\text{\# of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right) \\
&= \det\left(L_{\sim 1,\sim 1}\right) \qquad \left(\text{by Theorem 5.14.7, applied to } D = K_n^{\text{bidir}} \text{ and } r = 1\right) \\
&= \det \underbrace{\begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}}_{\text{an } (n-1)\times(n-1)\text{-matrix}}.
\end{aligned}$$

How do we compute this determinant? Here are three ways:

- The most elementary approach is using row transformations:

$$\begin{aligned}
&\det \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix} \\
&= \det \begin{pmatrix} n-1 & -1 & -1 & -1 & \cdots & -1 \\ -n & n & 0 & 0 & \cdots & 0 \\ -n & 0 & n & 0 & \cdots & 0 \\ -n & 0 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -n & 0 & 0 & 0 & \cdots & n \end{pmatrix} \qquad \left(\begin{array}{c}\text{here, we have} \\ \text{subtracted the 1st row} \\ \text{from each other row}\end{array}\right) \\
&= n^{n-2} \det \begin{pmatrix} n-1 & -1 & -1 & -1 & \cdots & -1 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \qquad \left(\begin{array}{c}\text{here, we have} \\ \text{factored out} \\ \text{an } n \text{ from each} \\ \text{row except for} \\ \text{the first row}\end{array}\right) \\
&= n^{n-2} \underbrace{\det \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}}_{\substack{=1 \\ \text{(since the matrix is triangular} \\ \text{with diagonal entries 1,1,...,1)}}} \qquad \left(\begin{array}{c}\text{here, we have added the 2nd,} \\ \text{3rd, etc. rows to the 1st row}\end{array}\right) \\
&= n^{n-2}.
\end{aligned}$$

- 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
$$\begin{aligned}&\begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n \end{pmatrix}}_{=A} + \underbrace{\begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}}_{=u} \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}}_{=v}.\end{aligned}$$

- Here is an approach that is heavier on linear algebra (specifically, eigenvectors and eigenvalues[60]):
Let $(e_1, e_2, \ldots, e_{n-1})$ be the standard basis of the $\mathbb{R}$-vector space $\mathbb{R}^{n-1}$ (so that e_i is the column vector with its i-th coordinate equal to 1 and all its other coordinates equal to 0). Then, we can find the following $n-1$ eigenvectors of our $(n-1) \times (n-1)$-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 $n-2$ eigenvectors $e_1 - e_i$ for all $i \in \{2, 3, \ldots, n-1\}$, each of them with eigenvalue n (check this!);
  - the eigenvector $e_1 + e_2 + \cdots + e_{n-1}$ with eigenvalue 1 (check this!).

  Since these $n-1$ eigenvectors are linearly independent (check this!), they form a basis of $\mathbb{R}^{n-1}$. Hence, our matrix is similar to the diagonal matrix with diagonal entries $\underbrace{n, n, \ldots, n}_{n-2 \text{ times}}, 1$ (by [Treil17, Chapter 4, Theorem 2.1]), and therefore has determinant $\underbrace{nn \cdots n}_{n-2 \text{ times}} 1 = n^{n-2}$.

There are other ways as well. Either way, the result we obtain is n^{n-2}. Thus, we have proved (relying on the Matrix-Tree Theorem, which we haven't yet proved):

[60] See [Treil17, Chapter 4] for a refresher.

Theorem 5.14.8 (Cayley's formula)**.** Let n be a positive integer. Then, the # of spanning trees of the complete graph K_n is n^{n-2}.

In other words:

Corollary 5.14.9. 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}.

Proof. This is just Theorem 5.14.8, since the simple graphs with vertex set $\{1, 2, \ldots, n\}$ that are trees are precisely the spanning trees of K_n. $\square$

There are many ways to prove Cayley's formula (Theorem 5.14.8). I can particularly recommend the two combinatorial proofs given in [Galvin21, §2.4 and §2.5], as well as Joyal's proof sketched in [Leinst19]. Most textbooks on enumerative combinatorics give one proof or another; e.g., [Stanle18, Appendix to Chapter 9] gives three. Cayley's formula also appears in Aigner's and Ziegler's best-of compilation of mathematical proofs [AigZie18, Chapter 33] with four different proofs. Note that some of the sources use a matrix-tree theorem for **undirected** graphs; this is a particular case of our matrix-tree theorem.[61]

However, in order to complete our proof, we still need to prove the Matrix-Tree Theorem.

5.14.6. Preparations for the proof

In order to prepare for the proof of the Matrix-Tree Theorem, we state a simple lemma (yet another criterion for a digraph to be an arborescence):

Lemma 5.14.10. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that D has no cycles. Assume moreover that D has no arcs with source r. Assume furthermore that each vertex $v \in V \setminus \{r\}$ has outdegree 1. Then, the digraph D is an arborescence rooted to r.

This lemma is precisely Exercise 4.4 **(b)**, at least after reversing all arcs. But let us give a self-contained proof here:

[61] One more **remark:** In Corollary 5.14.9, we have counted the trees with n vertices (i.e., simple graphs with vertex set $\{1, 2, \ldots, n\}$ that are trees). It sounds equally natural to count the "unlabelled trees with n vertices", i.e., the equivalence classes of such trees up to isomorphism. Unfortunately, this is one of those "messy numbers" with no good expression: the best formula known is recursive. There is also an asymptotic formula ("Otter's formula", [Otter48]): the number of equivalence classes of n-vertex trees (up to isomorphism) is

$$\approx \beta \frac{\alpha^n}{n^{5/2}} \qquad \text{with } \alpha \approx 2.955 \text{ and } \beta \approx 0.5349.$$

Proof of Lemma 5.14.10. Let u be any vertex of D. Let $\mathbf{p} = (v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ be a longest path of D that starts at u. [62] Thus, $v_0 = u$.

We shall show that $v_k = r$. Indeed, assume the contrary. Thus, $v_k \neq r$, so that $v_k \in V \setminus \{r\}$. Hence, the vertex v_k has outdegree 1 (since we assumed that each vertex $v \in V \setminus \{r\}$ has outdegree 1). Thus, there exists an arc b of D that has source v_k. Consider this arc b, and let w be its target. Thus, appending the arc b and the vertex w to the end of the path $\mathbf{p}$, we obtain a walk

$$\mathbf{w} = (v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k, b, w)$$

of D that starts at u (since $v_0 = u$). Proposition 4.5.9 shows that this walk $\mathbf{w}$ either is a path or contains a cycle. Hence, $\mathbf{w}$ is a path (since D has no cycles). Thus, $\mathbf{w}$ is a path of D that starts at u. Since $\mathbf{w}$ is longer than $\mathbf{p}$ (namely, longer by 1), this shows that $\mathbf{p}$ is not the longest path of D that starts at u. But this contradicts the very definition of $\mathbf{p}$.

This contradiction shows that our assumption was false. Hence, $v_k = r$. Thus, $\mathbf{p}$ is a path from u to r (since $v_0 = u$ and $v_k = r$). Therefore, the digraph D has a path from u to r (namely, $\mathbf{p}$).

Forget that we fixed u. We thus have shown that for each vertex u of D, the digraph D has a path from u to r. In other words, r is a to-root of D. Furthermore, we have $\deg^+ r = 0$ (since D has no arcs with source r), and each $v \in V \setminus \{r\}$ satisfies $\deg^+ v = 1$ (since we have assumed that each vertex $v \in V \setminus \{r\}$ has outdegree 1). In other words, the digraph D satisfies Statement A'6 from the dual arborescence equivalence theorem (Theorem 5.10.5). Therefore, it satisfies Statement A'1 from that theorem as well (since all six statements A'1, A'2, ..., A'6 are equivalent). In other words, D is an arborescence rooted to r. This proves Lemma 5.14.10. $\square$

5.14.7. The Matrix-Tree Theorem: proof

We shall now prove the Matrix-Tree Theorem (Theorem 5.14.7), guided by the following battle plan:

1. First, we will prove it in the case when each vertex $v \in V \setminus \{r\}$ has outdegree 1. In this case, after removing all arcs with source r from D (these arcs do not matter, since neither the submatrix $D_{\sim r, \sim r}$ nor the spanning arborescences rooted to r depend on them), we have essentially two options (subcases): either D is itself an arborescence or D has a cycle.

2. Then, we will prove the matrix-tree theorem in the slightly more general case when each $v \in V \setminus \{r\}$ has outdegree ≤ 1. This is easy, since a vertex $v \in V \setminus \{r\}$ having outdegree 0 trivializes the theorem.

[62] Such a path clearly exists, since the length-0 path (u) is a path of D that starts at u, and since a path of D cannot have length larger than $|V| - 1$.

3. Finally, we will prove the theorem in the general case. This is done by strong induction on the number 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 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 blue arcs (call it D^{red}) and the subdigraph of D obtained by removing all red arcs (call it D^{blue}). You can then apply the induction hypothesis to D^{red} and to D^{blue} (since each of these two subdigraphs has fewer arcs than D), and add the results together. The good news is that both the # of spanning arborescences rooted to r and the determinant $\det(L_{\sim r,\sim r})$ "behave additively" (we will soon see what this means).

So let us begin with Step 1. We first study a very special case:

Lemma 5.14.11. Let $D = (V, A, \psi)$ be a multidigraph. Let r be a vertex of D. Assume that D has no cycles. Assume moreover that D has no arcs with source r. Assume furthermore 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) Assume that $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. Let L be the Laplacian of D. Then, $\det(L_{\sim r,\sim r}) = 1$.

Proof. **(a)** Lemma 5.14.10 shows that the digraph D itself is an arborescence rooted to r.

As a consequence, D itself is a spanning arborescence of D rooted to r.

Therefore, $|A| = |V| - 1$ (by Statement A'2 in the Dual Arborescence Equivalence Theorem (Theorem 5.10.5)[63]). Hence, D has no spanning arborescences other than itself (because the condition $|A| = |V| - 1$ would get destroyed as soon as we remove an arc). So the only spanning arborescence of D rooted to r is D itself. This proves Lemma 5.14.11 **(a)**.

(b) We WLOG assume that $r = n$ (otherwise, we can swap r with n, so that $L_{\sim r,\sim r}$ becomes $L_{\sim n,\sim n}$).

Let D' be the digraph D with a loop added at each vertex – i.e., the multidigraph obtained from D by adding n extra arcs $\ell_1, \ell_2, \ldots, \ell_n$ and letting each arc ℓ_i have source i and target i.

Let S_{n-1} denote the group of permutations of the set

$$\{1, 2, \ldots, n-1\} = \underbrace{\{1, 2, \ldots, n\}}_{=V} \setminus \left\{ \underbrace{n}_{=r} \right\} = V \setminus \{r\}.$$

[63] or by the fact that $|A|$ is the sum of the outdegrees of all vertices of D

Now, from $r = n$, we have

$$\det\left(L_{\sim r,\sim r}\right) = \det\left(L_{\sim n,\sim n}\right) = \sum_{\sigma \in S_{n-1}} \operatorname{sign} \sigma \cdot \prod_{i=1}^{n-1} L_{i,\sigma(i)} \tag{22}$$

(by the Leibniz formula for the determinant). We shall now study the addends in the sum on the right hand side of this equality. Specifically, we will show that the only addend whose product $\prod_{i=1}^{n-1} L_{i,\sigma(i)}$ is nonzero is the addend for $\sigma = \operatorname{id}$.

Indeed, let $\sigma \in S_{n-1}$ be a permutation such that the product $\prod_{i=1}^{n-1} L_{i,\sigma(i)}$ is nonzero. We shall prove that $\sigma = \operatorname{id}$.

Consider an arbitrary $v \in \{1, 2, \ldots, n-1\}$. Then, $L_{v,\sigma(v)} \neq 0$ (because $L_{v,\sigma(v)}$ is a factor in the product $\prod_{i=1}^{n-1} L_{i,\sigma(i)}$, which is nonzero). However, the definition of L yields $L_{v,\sigma(v)} = \left(\deg^+ v\right) \cdot [v = \sigma(v)] - a_{v,\sigma(v)}$. Thus,

$$\left(\deg^+ v\right) \cdot [v = \sigma(v)] - a_{v,\sigma(v)} = L_{v,\sigma(v)} \neq 0.$$

Hence, at least one of the numbers $[v = \sigma(v)]$ and $a_{v,\sigma(v)}$ is nonzero. In other words, we have $v = \sigma(v)$ (this is what it means for $[v = \sigma(v)]$ to be nonzero) or the digraph D has an arc with source v and target $\sigma(v)$ (because this is what it means for $a_{v,\sigma(v)}$ to be nonzero). In either case, the digraph D' has an arc with source v and target $\sigma(v)$ (because if $v = \sigma(v)$, then one of the loops we added to D does the trick). We can apply the same argument to $\sigma(v)$ instead of v, and obtain an arc with source $\sigma(v)$ and target $\sigma(\sigma(v))$. Similarly, we obtain an arc with source $\sigma(\sigma(v))$ and target $\sigma(\sigma(\sigma(v)))$. We can continue this reasoning indefinitely. By continuing it for n steps, we obtain a walk

$$\left(v, *, \sigma(v), *, \sigma^2(v), *, \sigma^3(v), \ldots, *, \sigma^n(v)\right)$$

in the digraph D', where each asterisk means an arc (we don't care about what these arcs are, so we are not giving them names). This walk cannot be a path (since it has $n+1$ vertices, but D' has only n vertices); thus, it must contain a cycle (by Proposition 4.5.9). All arcs of this cycle must be loops (because otherwise, we could remove the loops from this cycle and obtain a cycle of D, but we know that D has no cycles). In particular, its first arc is a loop. Thus, our above walk $\left(v, *, \sigma(v), *, \sigma^2(v), *, \sigma^3(v), \ldots, *, \sigma^n(v)\right)$ contains a loop (since the arcs of the cycle come from this walk). In other words, we have $\sigma^i(v) = \sigma^{i+1}(v)$ for some $i \in \{0, 1, \ldots, n-1\}$. Since σ is injective, we can apply σ^{-i} to both sides of this equality, and conclude that $v = \sigma(v)$. In other words, $\sigma(v) = v$.

Forget that we fixed v. We thus have shown that $\sigma(v) = v$ for each $v \in \{1, 2, \ldots, n-1\}$. In other words, $\sigma = \operatorname{id}$.

Forget that we fixed σ. We thus have proved that $\sigma = \mathrm{id}$ for each permutation $\sigma \in S_{n-1}$ for which the product $\prod\limits_{i=1}^{n-1} L_{i,\sigma(i)}$ is nonzero. In other words, the only permutation $\sigma \in S_{n-1}$ for which the product $\prod\limits_{i=1}^{n-1} L_{i,\sigma(i)}$ is nonzero is the permutation id.

Thus, the only nonzero addend on the right hand side of (22) is the addend corresponding to $\sigma = \mathrm{id}$. Hence, (22) can be simplified as follows:

$$\det(L_{\sim n,\sim n}) = \underbrace{\operatorname{sign}(\mathrm{id})}_{=1} \cdot \prod_{i=1}^{n-1} L_{i,\mathrm{id}(i)} = \prod_{i=1}^{n-1} L_{i,\mathrm{id}(i)}.$$

Since each $i \in \{1, 2, \ldots, n-1\}$ satisfies

$$\begin{aligned} L_{i,\mathrm{id}(i)} = L_{i,i} &= \underbrace{\left(\deg^+ i\right)}_{\substack{=1 \\ \text{(since } i \text{ has outdegree 1} \\ \text{(because each vertex } v \in V \setminus \{r\} \text{ has} \\ \text{outdegree 1, and we can apply this} \\ \text{to } v=i \text{ since } i \in \{1,2,\ldots,n-1\} = V \setminus \{r\}))}} \cdot \underbrace{[i=i]}_{=1} - \underbrace{a_{i,i}}_{\substack{=0 \\ \text{(since } D \text{ has no cycles} \\ \text{and thus cannot have} \\ \text{a loop with source } i)}} \qquad \text{(by the definition of } L) \\ &= 1 \cdot 1 - 0 = 1, \end{aligned}$$

this can be simplified to $\det(L_{\sim n,\sim n}) = \prod\limits_{i=1}^{n-1} 1 = 1$. This proves Lemma 5.14.11 **(b)**. ☐

Next, we drop the "no cycles" condition:

Lemma 5.14.12. 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. (Here and in the following, "**MTT**" is short for "Matrix-Tree Theorem", i.e., for Theorem 5.14.7.)

Proof. First of all, we note that an arc with source r cannot appear in any spanning arborescence of D rooted to r (since any such arborescence satisfies $\deg^+ r = 0$, according to Statement A'6 in the Dual Arborescence Equivalence Theorem (Theorem 5.10.5)). Furthermore, the arcs with source r do not affect the matrix $L_{\sim r,\sim r}$, since they only appear in the r-th row of the matrix L (but this r-th row is removed in $L_{\sim r,\sim r}$).

Hence, any arc with source r can be removed from D without disturbing anything we currently care about. Thus, we WLOG assume that D has no arcs with source r (else, we can just remove them from D).

We WLOG assume that $r = n$ (otherwise, we can swap r with n, so that $L_{\sim r,\sim r}$ becomes $L_{\sim n,\sim n}$).

We are in one of the following two cases:

Case 1: The digraph D has a cycle.

Case 2: The digraph D has no cycles.

Consider Case 1. In this case, D has a cycle $\mathbf{v} = (v_1, *, v_2, *, \ldots, *, v_m)$ (where we again are putting asterisks in place of the arcs). This cycle cannot contain r (since D has no arcs with source r). Thus, all its vertices $v_1, v_2, \ldots, v_m$ belong to $V \setminus \{r\}$. Hence, for each $i \in \{1, 2, \ldots, m-1\}$, the vertex v_i has outdegree 1 (since we assumed that each vertex $v \in V \setminus \{r\}$ has outdegree 1). Consequently, for each $i \in \{1, 2, \ldots, m-1\}$, the only arc of D that has source v_i is the arc that follows v_i on the cycle $\mathbf{v}$. Therefore, in the matrix L, the v_i-th row has a 1 in the v_i-th position (because $\deg^+(v_i) = 1$), a -1 in the v_{i+1}-th position (since the arc that follows v_i on the cycle $\mathbf{v}$ has source v_i and target v_{i+1}), and 0s in all other positions. Since $r = n$, the same must then be true for the matrix $L_{\sim r, \sim r}$: That is, the v_i-th row of the matrix $L_{\sim r, \sim r}$ has a 1 in the v_i-th position, a -1 in the v_{i+1}-th position, and 0s in all other positions. Thus, the sum of the v_1-th, v_2-th, ..., v_{m-1}-th rows of $L_{\sim r, \sim r}$ is the zero vector (since the 1s and the -1s just cancel out)[64].[65]

So we have found a nonempty set of rows of $L_{\sim r, \sim r}$ whose sum is the zero vector. This yields that the matrix $L_{\sim r, \sim r}$ is singular (by basic properties of determinants[66]), so its determinant is $\det(L_{\sim r, \sim r}) = 0$. On the other hand, the digraph D has no spanning arborescence (because, in order to get a spanning arborescence of D, we would have to remove at least one arc of our cycle $\mathbf{v}$

[64]Namely, the -1 in the v_{i+1}-th position of the v_i-th row gets cancelled by the 1 in the v_{i+1}-th position of the v_{i+1}-th row. (We are using the fact that $v_m = v_1$ here.)

[65]Let me illustrate this on a representative example: Assume that the numbers $v_1, v_2, \ldots, v_{m-1}, v_m$ are $1, 2, \ldots, m-1, 1$ (respectively). Then, the first $m-1$ rows of L look as follows:

$$\begin{matrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -1 \\ -1 & & & & & 1 \end{matrix}$$

(where all the missing entries are zeroes). Thus, the sum of these $m-1$ rows is the zero vector. The same is therefore true of the matrix $L_{\sim r, \sim r}$ (since the first $m-1$ rows of the latter matrix are just the first $m-1$ rows of L, with their r-th entries removed).

The general case is essentially the same as this example; the only difference is that the relevant rows are in other positions.

[66]Specifically, we are using the following fact: "Let M be a square matrix. If there is a certain nonempty set of rows of M whose sum is the zero vector, then the matrix M is singular.".

To prove this fact, we let S be this nonempty set. Choose one row from this set, and call it the **chosen row**. Now, add all the other rows from this set to this one chosen row. This operation does not change the determinant of M (since the determinant of a matrix is unchanged when we add one row to another), but the resulting matrix has a zero row (namely, the chosen row) and thus has determinant 0. Hence, the original matrix M must have had determinant 0 as well. In other words, M was singular, qed.

(since an arborescence cannot have a cycle); but then, the source of this arc would have outdegree 0, and thus we could no longer find a path from this source to r, so we would not obtain a spanning arborescence). In other words,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = 0.$$

Comparing this with $\det(L_{\sim r,\sim r}) = 0$, we conclude that the MTT holds in this case (since it claims that $0 = 0$). Thus, Case 1 is done.

Next, we consider Case 2. In this case, D has no cycles. Then, $\det(L_{\sim r,\sim r}) = 1$ (by Lemma 5.14.11 **(b)**) and

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = 1 \qquad (\text{by Lemma 5.14.11 } \textbf{(a)}).$$

Thus, the MTT boils down to $1 = 1$, which is again true.

So Lemma 5.14.12 is proved. □

Next, we venture into a mildly greater generality:

Lemma 5.14.13. 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 (= Matrix-Tree Theorem) holds for these D and r.

Proof. If each vertex $v \in V \setminus \{r\}$ has outdegree 1, then this is true by Lemma 5.14.12.

Thus, we WLOG assume that this is not the case. Hence, some vertex $v \in V \setminus \{r\}$ has outdegree $\neq 1$. Consider this v. The outdegree of v is $\neq 1$, but also ≤ 1 (by the hypothesis of the lemma). Hence, this outdegree must be 0. That is, there is no arc with source v.

WLOG assume that $r = n$ (otherwise, swap r with n).

We have $v \neq r$. Hence, the digraph D has no path from v to r (since any such path would include an arc with source v, but there is no arc with source v).

Therefore, D has no spanning arborescence rooted to r (because any such spanning arborescence would have to have a path from v to r). In other words,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = 0.$$

Also, $\det(L_{\sim r,\sim r}) = 0$ (since the v-th row of the matrix $L_{\sim r,\sim r}$ is 0 (because there is no arc with source v)). So the MTT boils down to $0 = 0$ again, and thus Lemma 5.14.13 is proved. □

We are now ready to prove the MTT in the general case:

Proof of Theorem 5.14.7. 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 remaining 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 a number such that $M \overset{j}{\equiv} N$, then

$$\det\left(M \overset{j}{+} N\right) = \det M + \det N.$$

Now, let us prove the MTT. We proceed by strong induction on the # of arcs of D.

Induction step: Let $m \in \mathbb{N}$. Assume (as the induction hypothesis) that the MTT holds for all digraphs D that have $< m$ arcs. We must now prove it for our digraph D with m arcs.

WLOG assume that $r = n$ (otherwise, swap r with n).

If each vertex $v \in V \setminus \{r\}$ has outdegree ≤ 1, then the MTT holds by Lemma 5.14.13. Thus, we WLOG assume that some vertex $v \in V \setminus \{r\}$ has outdegree > 1. Pick such a vertex v. We color each arc with source v either red or blue, making sure that at least one arc is red and at least one arc is blue. (We can do this, since v has outdegree > 1.) All arcs that do not have source v remain uncolored.

Now, let D^{red} be the subdigraph obtained from D by removing all blue arcs. Then, D^{red} has fewer arcs than D. In other words, D^{red} has $< m$ arcs. Hence, the induction hypothesis yields that the MTT holds for D^{red}. That is, we have

$$\left(\text{\# of spanning arborescences of } D^{\text{red}} \text{ rooted to } r\right) = \det\left(L^{\text{red}}_{\sim r, \sim r}\right),$$

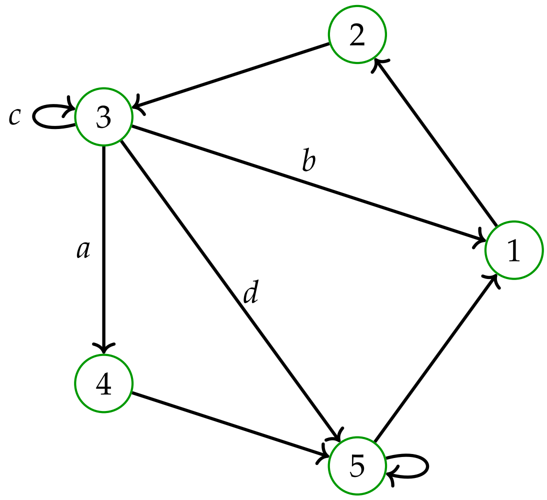
where L^{red} means the Laplacian of D^{red}.

Likewise, let D^{blue} be the subdigraph obtained from D by removing all red arcs. Then, D^{blue} has fewer arcs than D. Hence, the induction hypothesis yields that the MTT holds for D^{blue}. That is,

$$\left(\text{\# of spanning arborescences of } D^{\text{blue}} \text{ rooted to } r\right) = \det\left(L^{\text{blue}}_{\sim r, \sim r}\right),$$

where L^{blue} means the Laplacian of D^{blue}.

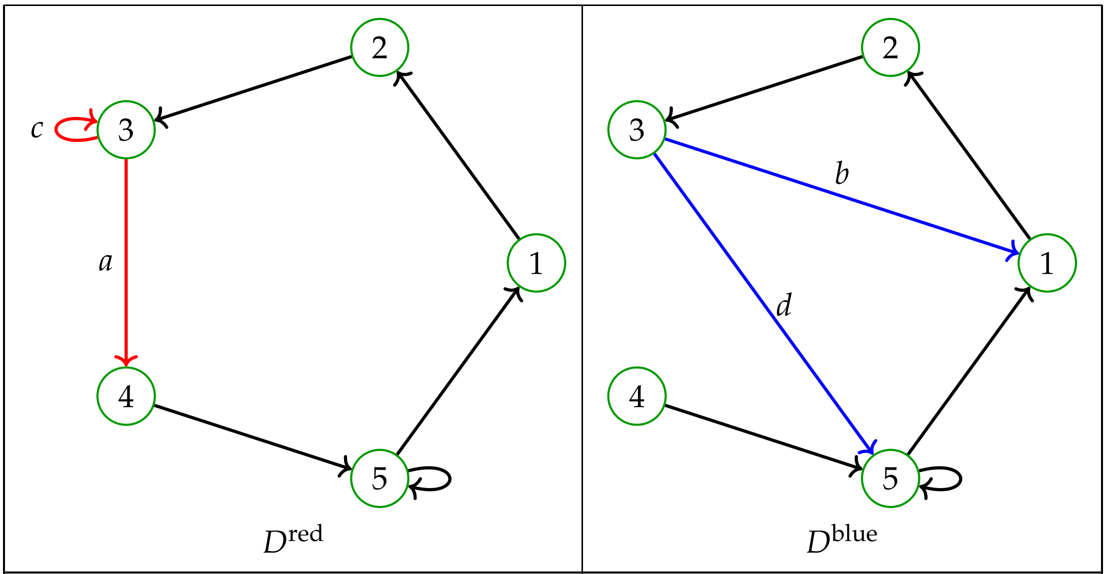
Example 5.14.14. Let D be the multidigraph



with $r = 1$. Its Laplacian is

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Let us pick $v = 3$ (this is a vertex with outdegree > 1), and let us color the arcs a and c red and the arcs b and d blue (various other options are possible). Then, D^{red} and D^{blue} look as follows (along with their Laplacians L^{red} and L^{blue}):



$$L^{\text{red}} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad L^{\text{blue}} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Now, the digraphs D, D^{blue} and D^{red} differ only in the arcs with source v, and as far as the latter arcs are concerned, the arcs of D are divided between D^{blue} and D^{red}. Hence, by the definition of the Laplacian, we have

$$L^{\text{red}} \overset{v}{\equiv} L^{\text{blue}} \qquad \text{and} \qquad L^{\text{red}} \overset{v}{+} L^{\text{blue}} = L.$$

Thus,

$$L^{\text{red}}_{\sim r,\sim r} \overset{v}{\equiv} L^{\text{blue}}_{\sim r,\sim r} \qquad \text{and} \qquad L^{\text{red}}_{\sim r,\sim r} \overset{v}{+} L^{\text{blue}}_{\sim r,\sim r} = L_{\sim r,\sim r}$$

(here, we have used the fact that $r = n$ and $v \neq r$, so that when we remove the r-th row and the r-th column of the matrix L, the v-th row remains the v-th row). Hence,

$$\det \left(\underbrace{L_{\sim r,\sim r}}_{= L^{\text{red}}_{\sim r,\sim r} \overset{v}{+} L^{\text{blue}}_{\sim r,\sim r}} \right) = \det \left(L^{\text{red}}_{\sim r,\sim r} \overset{v}{+} L^{\text{blue}}_{\sim r,\sim r} \right) = \det \left(L^{\text{red}}_{\sim r,\sim r} \right) + \det \left(L^{\text{blue}}_{\sim r,\sim r} \right)$$

(by the multilinearity of the determinant).

However, a similar equality holds for the # of spanning arborescences: namely, we have

$$\begin{aligned}
&(\text{\# of spanning arborescences of } D \text{ rooted to } r) \\
&= \left(\text{\# of spanning arborescences of } D^{\text{red}} \text{ rooted to } r\right) \\
&\qquad + \left(\text{\# of spanning arborescences of } D^{\text{blue}} \text{ rooted to } r\right).
\end{aligned}$$

Here is why: Recall that an arborescence rooted to r must satisfy $\deg^+ v = 1$ (by Statement A'6 in the Dual Arborescence Equivalence Theorem (Theorem 5.10.5), since $v \in V \setminus \{r\}$). In other words, an arborescence rooted to r must contain exactly one arc with source v. In particular, a spanning arborescence of D rooted to r must contain either a red arc or a blue arc, but not both at the same time. In the former case, it is a spanning arborescence of D^{red}; in the latter, it is a spanning arborescence of D^{blue}. Conversely, any spanning arborescence of D^{red} or of D^{blue} rooted to r is automatically a spanning arborescence of D rooted to r. Thus,

$$\begin{aligned}
&(\text{\# of spanning arborescences of } D \text{ rooted to } r) \\
&= \underbrace{\left(\text{\# of spanning arborescences of } D^{\text{red}} \text{ rooted to } r\right)}_{\substack{=\det\left(L^{\text{red}}_{\sim r,\sim r}\right) \\ \text{(as we saw above)}}} \\
&\qquad + \underbrace{\left(\text{\# of spanning arborescences of } D^{\text{blue}} \text{ rooted to } r\right)}_{\substack{=\det\left(L^{\text{blue}}_{\sim r,\sim r}\right) \\ \text{(as we saw above)}}} \\
&= \det\left(L^{\text{red}}_{\sim r,\sim r}\right) + \det\left(L^{\text{blue}}_{\sim r,\sim r}\right) = \det\left(L_{\sim r,\sim r}\right)
\end{aligned}$$

(since we proved that $\det\left(L_{\sim r,\sim r}\right) = \det\left(L^{\text{red}}_{\sim r,\sim r}\right) + \det\left(L^{\text{blue}}_{\sim r,\sim r}\right)$). That is, the MTT holds for our digraph D and its vertex r. This completes the induction step, and thus the MTT (Theorem 5.14.7) is proved. $\square$

Our above proof of Theorem 5.14.7 has followed [Stanle18, Theorem 10.4]. Other proofs can be found across the literature, e.g., in [VanEhr51, Theorem 7], in [Margol10, Theorem 2.8], in [DeLeen19, Theorem 1] and in [Holzer22, Theorem 2.5.3]. (Some of these sources prove more general versions of the theorem. Confusingly, each source uses different notations and works in a slightly different setup, although most of them quickly reveal themselves to be equivalent upon some introspection.)

5.14.8. Further exercises on the Laplacian

Exercise 5.18. Let $G = (V, E, \varphi)$ be a multigraph. Let L be the Laplacian of the digraph G^{bidir}. Prove that L is positive semidefinite.

[**Hint:** Write L as $N^T N$, where N or N^T is some matrix you have seen before.

Note that the statement is not true if we replace G^{bidir} by an arbitrary digraph D.]

The following two exercises stand at the beginning of the theory of **chip-firing** and related dynamical systems on a digraph (see [CorPer18], [Klivan19] and [JoyMel17] for much more). While the Laplacian is not mentioned in them directly, it is implicitly involved in the definition of a "donation" (how?).

Exercise 5.19. Let $D = (V, A, \psi)$ be a strongly connected multidigraph.

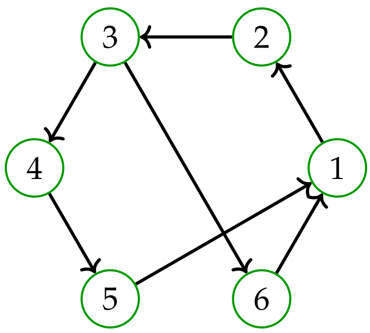
A **wealth distribution** on D shall mean a family $(k_v)_{v \in V}$ of integers (one for each vertex $v \in V$). If $k = (k_v)_{v \in V}$ is a wealth distribution, then we refer to each value k_v as the **wealth** of the vertex v, and we define the **total wealth** of k to be the sum $\sum\limits_{v \in V} k_v$. We say that a vertex v is **in debt** in a given wealth distribution $k = (k_v)_{v \in V}$ if its wealth k_v is negative.

For any vertices v and w, we let $a_{v,w}$ denote the number of arcs that have source v and w.

A **donation** is an operation that transforms a wealth distribution as follows: We choose a vertex v, and we decrease its wealth by its outdegree $\deg^+ v$, and then increase the wealth of each vertex $w \in V$ (including v itself) by $a_{v,w}$. (You can think of v as donating a unit of wealth for each arc that has source v. This unit flows to the target to this arc. Note that a donation does not change the total wealth.)

Let k be a wealth distribution on D whose total wealth is larger than $|A| - |V|$. Prove that by an appropriately chosen finite sequence of donations, we can ensure that no vertex is in debt.

[**Example:** For instance, consider the digraph



with wealth distribution $(k_1, k_2, k_3, k_4, k_5, k_6) = (-1, -1, 1, 2, 0, 1)$. The vertices 1 and 2 are in debt here, but it is possible to get all vertices out of debt by having the vertices $4, 5, 6, 1$ donate in some order (the order clearly does not matter for the result[67]).

Note that vertices are allowed to donate multiple times (although in the above example, this was unnecessary).]

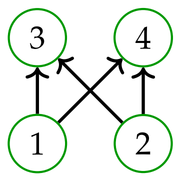
[**Hint:** A donation will be called **safe** if its donor v (that is, the vertex chosen to lose wealth) satisfies $k_v \geq \deg^+ v$, where k is the wealth distribution just before this donation. Start by showing that if the total wealth is larger than $|A| - |V|$, then at least one vertex v has wealth $\geq \deg^+ v$ (and thus can make a safe donation). Next, show that for any given wealth distribution k, there are only finitely many wealth distributions that can be obtained from k by a sequence of safe donations. Finally, for any vertex v, find a rational quantity that increases every time that a donor distinct from v makes a donation. Conclude that in a sufficiently long sequence of safe donations, every vertex must appear as a donor. But a donor of a safe donation must be out of debt just before its safe donation, and will never go back into debt.]

Exercise 5.20. We continue with the setting and terminology of Exercise 5.19.

A **clawback** is an operation that transforms a wealth distribution as follows: We choose a vertex v, and we increase its wealth by its outdegree $\deg^+ v$, and then decrease the wealth of each vertex $w \in V$ (including v itself) by $a_{v,w}$. (Thus, a clawback is the inverse of a donation.)

Let k be a wealth distribution on D whose total wealth is larger than $|A| - |V|$. Prove that by an appropriately chosen finite sequence of clawbacks, we can ensure that no vertex is in debt.

[**Remark:** Note that we are still assuming D to be strongly connected. Otherwise, the truth of the claim is not guaranteed. For instance, for the digraph



with wealth distribution $(k_1, k_2, k_3, k_4) = (0, 0, -1, 2)$, no sequence of donations and clawbacks will result in every vertex being out of debt (since the wealth difference $k_4 - k_3$ is preserved under any donation or clawback, but this difference is too large to come from a debt-free distribution with total weight 1).]

[**Hint:** Show that any donation is equivalent to an appropriately chosen composition of clawbacks. Something we know about the Laplacian may come useful here.]

[67] Depending on the order, some vertices will go into debt in the process, but this is okay as long as they ultimately end up debt-free.

5.14.9. Application: Counting Eulerian circuits of K_n^{bidir}

Here is one more consequence of the MTT:

Proposition 5.14.15. 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. Let r be the source of the arc a. The digraph K_n^{bidir} is balanced, and each of its vertices has outdegree $n-1$. By the BEST′ theorem (Theorem 5.10.4), we have

$$\left(\text{\# of Eulerian circuits of } K_n^{\text{bidir}} \text{ whose first arc is } a\right)$$

$$= \underbrace{\left(\text{\# of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } r\right)}_{\substack{=n^{n-2} \\ \text{(as we saw in Subsection 5.14.5 in the case when } r=1, \\ \text{and can similarly prove for arbitrary } r)}} \cdot \prod_{u=1}^{n} \left(\underbrace{\deg^+ u}_{=n-1} - 1 \right)!$$

$$= n^{n-2} \cdot \prod_{u=1}^{n} (n-2)! = n^{n-2} \cdot (n-2)!^n,$$

qed. □

In comparison, there is no good formula known for the # of Eulerian circuits of the undirected graph K_n. For n even, this # is 0 of course (since K_n has vertices of odd degree in this case). For n odd, the # grows very fast, but little else is known about it (see `https://oeis.org/A135388` for some known values, and see Exercise 5.22 for a divisibility property).

Exercise 5.21. Let n be a positive integer. Let $N = \{1, 2, \ldots, n\}$. A map $f : N \to N$ is said to be ***n*-potent** if each $i \in N$ satisfies $f^{n-1}(i) = n$. (As usual, f^k denotes the k-fold composition $f \circ f \circ \cdots \circ f$.)

Prove that the # of n-potent maps $f : N \to N$ is n^{n-2}.

[**Hint:** What do these n-potent maps have to do with trees?]

Exercise 5.22. Let $n = 2m + 1 > 2$ be an odd integer. Let e be an edge of the (undirected) complete graph K_n. Prove that the # of Eulerian circuits of K_n that start with e is a multiple of $(m-1)!^n$.

[**Hint:** Argue that each Eulerian circuit of K_n is an Eulerian circuit of a unique balanced tournament. Here, a "balanced tournament" means a balanced digraph obtained from K_n by orienting each edge.]

5.15. The undirected Matrix-Tree Theorem

5.15.1. The theorem

The Matrix-Tree Theorem becomes simpler if we apply it to a digraph of the form G^{bidir}:

Theorem 5.15.1 (undirected Matrix-Tree Theorem)**.** 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 of G that have endpoints i and j (with loops counting 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 $\det(tI_n + L)$ (here, I_n denotes 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 $c_0, c_1, \ldots, c_n$ are numbers. (Note that this is the characteristic polynomial of L up to substituting $-t$ for t and multiplying by a power of -1. Some of its coefficients are $c_n = 1$ and $c_{n-1} = \operatorname{Tr} L$ and $c_0 = \det L$.) 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 know that 0 is an eigenvalue of L, 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, Proposition 5.13.1 **(b)** shows 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, we have

$$\begin{aligned}
&(\text{\# of spanning trees of } G) \\
&= \left(\text{\# of spanning arborescences of } G^{\text{bidir}} \text{ rooted to } r\right) \\
&= \det\left(L_{\sim r,\sim r}\right) \qquad \text{(by the Matrix-Tree Theorem (Theorem 5.14.7))}.
\end{aligned}$$

This proves Theorem 5.15.1 **(a)**.

(b) We claim that

$$c_1 = \sum_{r=1}^{n} \det\left(L_{\sim r,\sim r}\right). \tag{23}$$

Note that this is a purely linear-algebraic result, and has nothing to do with the fact that L is the Laplacian of a digraph; it holds just as well if L is replaced by any square matrix.

Once (23) is proved, Theorem 5.15.1 **(b)** will easily follow, because (23) entails

$$\begin{aligned}
\frac{1}{n} c_1 &= \frac{1}{n} \sum_{r=1}^{n} \underbrace{\det\left(L_{\sim r,\sim r}\right)}_{\substack{=(\text{\# of spanning trees of } G) \\ \text{(by Theorem 5.15.1 \textbf{(a)})}}} = \frac{1}{n} \underbrace{\sum_{r=1}^{n} (\text{\# of spanning trees of } G)}_{=n\cdot(\text{\# of spanning trees of } G)} \\
&= \frac{1}{n} \cdot n\,(\text{\# of spanning trees of } G) = (\text{\# of spanning trees of } G).
\end{aligned}$$

Thus, it remains to prove (23).

A rigorous proof of (23) can be found in [21s, Proposition 6.4.29] or in `https://math.stackexchange.com/a/3989575/` (both of these references actually describe all coefficients $c_0, c_1, \ldots, c_n$ of the polynomial $\det(tI_n + L)$, not just the t^1-coefficient c_1). We shall merely outline the proof of (23) on a convenient example. We want to compute c_1. In other words, we want to compute the coefficient of t^1 in the polynomial $\det(tI_n + L)$ (since c_1 is defined to be this very coefficient). Let us say that $n = 4$, so that L has the form

$$L = \begin{pmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \\ a''' & b''' & c''' & d''' \end{pmatrix}.$$

Thus,

$$\det\left(tI_n + L\right) = \det \begin{pmatrix} t+a & b & c & d \\ a' & t+b' & c' & d' \\ a'' & b'' & t+c'' & d'' \\ a''' & b''' & c''' & t+d''' \end{pmatrix}.$$

Imagine expanding the right hand side (using the Leibniz formula) and expanding the resulting products further. For instance, the product

$$(t+a)\left(t+b'\right) d'' c'''$$

becomes $ttd''c''' + tb'd''c''' + atd''c''' + ab'd''c'''$. In the huge sum that results, we are interested in those addends that contain exactly one t, because it is precisely these addends that contribute to the coefficient of t^1 in the polynomial $\det(tI_n + L)$. Where do these addends come from? To pick up exactly one t from a product like $(t+a)(t+b')d''c'''$, we need to have at least one diagonal entry in our product (for example, we cannot pick up any t from the product $cd'b''a'''$), and we need to pick out the t from this diagonal entry (rather than, e.g., the a or b' or c'' or d'''). If we pick the r-th diagonal entry, then the rest of the product is part of the expansion of $\det(L_{\sim r,\sim r})$ (since we must not pick any further ts and thus can pretend that they are not there in the first place). Thus, the total t^1-coefficient in $\det(tI_n + L)$ will be $\sum\limits_{r=1}^{n} \det(L_{\sim r,\sim r})$. This proves (23), and thus the proof of Theorem 5.15.1 **(b)** is complete.

(c) Consider the polynomial $\det(tI_n + L)$ introduced in part **(b)**, and in particular its t^1-coefficient c_1.

It is known that the characteristic polynomial $\det(tI_n - L)$ of L is a monic polynomial of degree n, and that its roots are the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of L. Hence, it can be factored as follows:

$$\det(tI_n - L) = (t - \lambda_1)(t - \lambda_2)\cdots(t - \lambda_n).$$

Substituting $-t$ for t on both sides of this equality, we obtain

$$\det(-tI_n - L) = (-t - \lambda_1)(-t - \lambda_2)\cdots(-t - \lambda_n).$$

Multiplying both sides of this equality by $(-1)^n$, we find

$$\begin{aligned}\det(tI_n + L) &= (t + \lambda_1)(t + \lambda_2)\cdots(t + \lambda_n) \\ &= (t + \lambda_1)(t + \lambda_2)\cdots(t + \lambda_{n-1})\, t \qquad (\text{since } \lambda_n = 0).\end{aligned}$$

Hence, the t^1-coefficient of the polynomial $\det(tI_n + L)$ is $\lambda_1\lambda_2\cdots\lambda_{n-1}$ (since this is clearly the t^1-coefficient on the right hand side). Since we defined c_1 to be the t^1-coefficient of the polynomial $\det(tI_n + L)$, we thus conclude that $c_1 = \lambda_1\lambda_2\cdots\lambda_{n-1}$. However, Theorem 5.15.1 **(b)** yields

$$(\text{\# of spanning trees of } G) = \frac{1}{n} \underbrace{c_1}_{=\lambda_1\lambda_2\cdots\lambda_{n-1}} = \frac{1}{n} \cdot \lambda_1\lambda_2\cdots\lambda_{n-1}.$$

This proves Theorem 5.15.1 **(c)**. □

5.15.2. Application: counting spanning trees of $K_{n,m}$

Laplacians of digraphs often have computable eigenvalues, so Theorem 5.15.1 **(c)** is actually pretty useful. A striking example of a # of spanning trees (specifically, of the n-hypercube graph Q_n, which we already met in Subsection 2.14.4) that can be counted using eigenvalues will appear in Exercise 5.26.

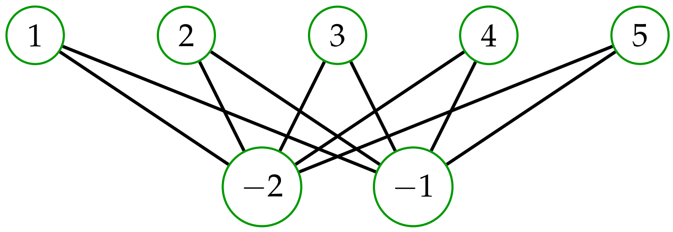
Here, however, let us give a simpler example, in which Theorem 5.15.1 **(a)** suffices:

Exercise 5.23. Let n and m be two positive integers. Let $K_{n,m}$ be the simple graph with $n+m$ vertices

$$1,2,\ldots,n \qquad \text{and} \qquad -1,-2,\ldots,-m,$$

where two vertices i and j are adjacent if and only 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).

[For example, here is how $K_{5,2}$ looks like:



.]

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 follows:

$$L = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where

- A is a diagonal $n\times n$-matrix whose all diagonal entries are equal to m (since there are no edges between positive vertices, and since each positive vertex has degree m);
- B is an $n\times m$-matrix whose all entries equal -1;
- C is an $m\times n$-matrix whose all entries equal -1;
- D is a diagonal $m\times m$-matrix whose all diagonal entries are equal to n.

For instance, if $n=3$ and $m=2$, then

$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 2 & 0 & -1 & -1 \\ 0 & 0 & 2 & -1 & -1 \\ -1 & -1 & -1 & 3 & 0 \\ -1 & -1 & -1 & 0 & 3 \end{pmatrix}.$$

Theorem 5.15.1 **(a)** yields

$$(\text{\# of spanning trees of } K_{n,m}) = \det(L_{\sim r,\sim r}) \qquad \text{for any vertex } r \text{ of } K_{n,m};$$

thus, we need to compute $\det(L_{\sim r,\sim r})$ for some vertex r. We let $r = 1$. Then, the submatrix $L_{\sim r,\sim r} = L_{\sim 1,\sim 1}$ of L again can be written in block-matrix notation as follows:

$$L_{\sim r,\sim r} = \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix}, \qquad (24)$$

where

- $\widetilde{A}$ is a diagonal $(n-1) \times (n-1)$-matrix, whose all diagonal entries are equal to m;
- $\widetilde{B}$ is an $(n-1) \times m$-matrix whose all entries equal -1;
- $\widetilde{C}$ is an $m \times (n-1)$-matrix whose all entries equal -1;
- D is a diagonal $m \times m$-matrix whose all diagonal entries are equal to n.

Fortunately, determinants of block matrices are often not hard to compute, at least when some of the blocks are invertible. For example, the Schur complement provides a neat formula. Our life here is even easier, since $\widetilde{A}$ and D are multiples of identity matrices: namely, $\widetilde{A} = mI_{n-1}$ and $D = nI_m$. We perform a "blockwise row transformation" on the block matrix $L_{\sim r,\sim r} = \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix}$, specifically subtracting 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" $\begin{pmatrix} \widetilde{C} & D \end{pmatrix}$ (yes, this is legitimate – it's the same as left-multiplying by the block matrix $\begin{pmatrix} I_{n-1} & 0 \\ -\widetilde{C}\widetilde{A}^{-1} & I_m \end{pmatrix}$, which has determinant 1 because it is lower-triangular). As a result, we obtain

$$\begin{aligned} \det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix} &= \det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} - \widetilde{C}\widetilde{A}^{-1}\widetilde{A} & D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \end{pmatrix} \\ &= \det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ 0 & D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \end{pmatrix}. \end{aligned}$$

The matrix on the right is "block-upper triangular", so its determinant factors as follows:[68]

$$\det \begin{pmatrix} \widetilde{A} & \widetilde{B} \\ 0 & D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \end{pmatrix} = \det \widetilde{A} \cdot \det \left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \right).$$

[68] We are using the fact that if a matrix is block-triangular (with all diagonal blocks being square matrices), then its determinant is the product of the determinants of its diagonal blocks. See, e.g., `https://math.stackexchange.com/a/1221066/` or [Grinbe20, Exercise 6.29] for a proof of this fact.

Of course, $\det \widetilde{A} = m^{n-1}$, since $\widetilde{A}$ is a diagonal matrix with $m, m, \ldots, m$ on the diagonal. Computing $\det \left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right)$ is a bit more complicated, but still doable: The matrix $\widetilde{A}^{-1}$ is a diagonal matrix with $m^{-1}, m^{-1}, \ldots, m^{-1}$ on the diagonal; thus, its role in the product $\widetilde{C}\widetilde{A}^{-1}\widetilde{B}$ is merely to multiply everything by m^{-1}. Hence, $\widetilde{C}\widetilde{A}^{-1}\widetilde{B} = m^{-1}\widetilde{C}\widetilde{B}$. Since all entries of $\widetilde{C}$ and $\widetilde{B}$ are -1's, we see that all entries of $\widetilde{C}\widetilde{B}$ are $(n-1)$'s. Putting all of this together, we see that $D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}$ is the $m \times m$-matrix whose all diagonal entries are equal to $n - m^{-1}(n-1)$ and whose all off-diagonal entries are equal to $-m^{-1}(n-1)$. We have already computed the determinant of a matrix much like this back in our proof of Cayley's Formula (Subsection 5.14.5); let us deal with the general case:

Proposition 5.15.2. Let $n \in \mathbb{N}$. Let x and a be two numbers. Then,

$$\det \underbrace{\begin{pmatrix} x & a & a & \cdots & a & a \\ a & x & a & \cdots & a & a \\ a & a & x & \cdots & a & a \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a & a & a & \cdots & x & a \\ a & a & a & \cdots & a & x \end{pmatrix}}_{\substack{\text{the } n\times n\text{-matrix} \\ \text{whose diagonal entries are } x \\ \text{and whose off-diagonal entries are } a}} = (x + (n-1)\,a)\,(x-a)^{n-1}.$$

Proposition 5.15.2 can be proved using similar reasoning as the determinant in Subsection 5.14.5; we will say more about it later. For now, let us apply it to m, $n - m^{-1}(n-1)$ and $-m^{-1}(n-1)$ instead of n, x and a, to obtain

$$\begin{aligned} \det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right) &= \underbrace{\left(\left(n - m^{-1}(n-1)\right) + (m-1)\left(-m^{-1}(n-1)\right)\right)}_{=1} \\ &\qquad \cdot \left(\underbrace{\left(n - m^{-1}(n-1)\right) - \left(-m^{-1}(n-1)\right)}_{=n}\right)^{m-1} \\ &= n^{m-1}. \end{aligned}$$

Now, it is time to combine everything we know. Theorem 5.15.1 **(a)** yields

$$\begin{aligned}
(\text{\# of spanning trees of } K_{n,m}) &= \det\left(L_{\sim r,\sim r}\right) \\
&= \det\begin{pmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & D \end{pmatrix} \qquad \text{(by (24))} \\
&= \det\begin{pmatrix} \widetilde{A} & \widetilde{B} \\ 0 & D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B} \end{pmatrix} \\
&= \underbrace{\det \widetilde{A}}_{=m^{n-1}} \cdot \underbrace{\det\left(D - \widetilde{C}\widetilde{A}^{-1}\widetilde{B}\right)}_{=n^{m-1}} \\
&= m^{n-1} \cdot n^{m-1}.
\end{aligned}$$

□

Thus, we have obtained the following:

Theorem 5.15.3. Let n and m be two positive integers. Let $K_{n,m}$ be the simple graph with $n+m$ vertices

$$1, 2, \ldots, n \qquad \text{and} \qquad -1, -2, \ldots, -m,$$

where two vertices i and j are adjacent if and only if they have opposite signs. Then,

$$(\text{\# of spanning trees of } K_{n,m}) = m^{n-1} \cdot n^{m-1}.$$

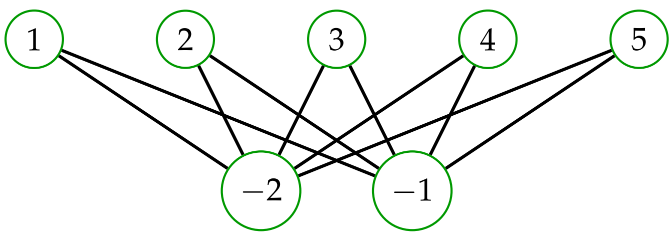
See [AbuSbe88] for a combinatorial proof of this theorem.

Exercise 5.24. Let n be a positive integer. Let $K_{n,2}$ be the simple graph with vertex set $\{1, 2, \ldots, n\} \cup \{-1, -2\}$ such that two vertices of $K_{n,2}$ are adjacent if and only 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). We regard $K_{n,2}$ as a multigraph in the usual way.

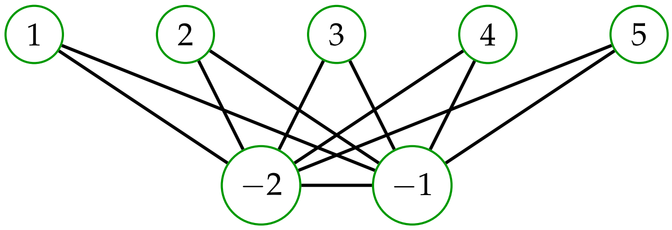
(a) Without using the matrix-tree theorem, prove that the number of spanning trees of $K_{n,2}$ is $n \cdot 2^{n-1}$.

(b) Let $K'_{n,2}$ be the graph obtained by adding a new edge $\{-1, -2\}$ to $K_{n,2}$. How many spanning trees does $K'_{n,2}$ have?

[**Example:** Here is the graph $K_{n,2}$ for $n = 5$:



And here is the corresponding graph $K'_{n,2}$:



]

Exercise 5.25. Let n be a positive integer. Let A be the $(n-1) \times (n-1)$-matrix

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{pmatrix},$$

whose (i,j)-th entry is

$$A_{i,j} := \begin{cases} 2, & \text{if } i = j; \\ -1, & \text{if } |i-j| = 1; \\ 0, & \text{otherwise} \end{cases} \qquad \text{for all } i,j \in \{1,2,\ldots,n-1\}.$$

Prove that $\det A = n$.

[**Hint:** Recall Example 5.4.4.]

Exercise 5.26. Let n be a positive integer. Let Q_n be the n-hypercube graph (as defined in Definition 2.14.7). Recall that its vertex set is the set $V := \{0,1\}^n$ of length-n bitstrings, and that two vertices are adjacent if and only if they differ in exactly one bit. Our goal is to compute the # of spanning trees of Q_n.

Let D be the digraph Q_n^{bidir}. Let L be the Laplacian of D. We regard L as a $V \times V$-matrix (i.e., as a $2^n \times 2^n$-matrix whose rows and columns are indexed by bitstrings in V).

We shall use the notation a_i for the i-th entry of a bitstring a. Thus, each bitstring $a \in V$ has the form $a = (a_1, a_2, \ldots, a_n)$. (We shall avoid the shorthand notation $a_1 a_2 \cdots a_n$ here, as it could be mistaken for an actual product.)

For any two bitstrings $a, b \in V$, we define the number $\langle a, b \rangle$ to be the integer $a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$.

(a) Prove that every bitstring $a \in V$ satisfies

$$\sum_{b \in V} (-1)^{\langle a,b \rangle} = \begin{cases} 2^n, & \text{if } a = \mathbf{0}; \\ 0, & \text{otherwise.} \end{cases}$$

Here, $\mathbf{0}$ denotes the bitstring $(0, 0, \ldots, 0) \in V$.

Now, define a further $V \times V$-matrix G by requiring that its (a, b)-th entry is

$$G_{a,b} = (-1)^{\langle a,b \rangle} \qquad \text{for any } a, b \in V.$$

Furthermore, define a diagonal $V \times V$-matrix D by requiring that its (a, a)-th entry is

$$\begin{aligned} D_{a,a} &= 2 \cdot (\# \text{ of } i \in \{1, 2, \ldots, n\} \text{ such that } a_i = 1) \\ &= 2 \cdot (\text{the number of 1s in } a) \qquad \text{for any } a \in V \end{aligned}$$

(and its off-diagonal entries are 0).

Prove the following:

(b) We have $G^2 = 2^n \cdot I$, where I is the identity $V \times V$-matrix.

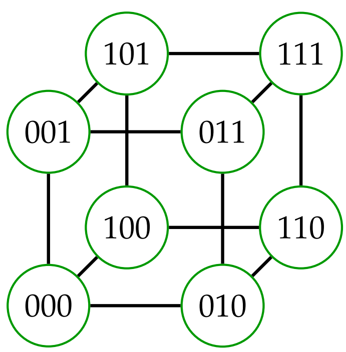
(c) We have $GLG^{-1} = D$.

(d) The eigenvalues of L are $2k$ for all $k \in \{0, 1, \ldots, n\}$, and each eigenvalue $2k$ appears with multiplicity $\dbinom{n}{k}$.

(e) The # of spanning trees of Q_n is

$$\frac{1}{2^n} \prod_{k=1}^{n} (2k)^{\binom{n}{k}}.$$

[**Example:** As an example, here is the case $n = 3$. In this case, the graph Q_n looks as follows:

$Q_3 =$ 

The matrices L, G and D are

$$L = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 3 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 3 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 3 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & -1 & 3 \end{pmatrix},$$

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix},$$

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix},$$

where the rows and the columns are ordered by listing the eight bitstrings $a \in V$ in the order $000, 001, 010, 011, 100, 101, 110, 111$.]

As we promised, let us make a few more remarks about Proposition 5.15.2. While this proposition can be proved by fairly straightforward row transformations (first subtracting the first row from all the other rows, then factoring an $x - a$ from all the latter rows, then subtracting a times each of the latter rows to the first row to obtain a triangular matrix), it can also be viewed as a particular case of either of the following two determinantal identities:

Proposition 5.15.4. Let $n \in \mathbb{N}$. Let $a_1, a_2, \ldots, a_n$ be n numbers, and let x be a further number. Then,

$$\det \underbrace{\begin{pmatrix} x & a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_1 & x & a_2 & \cdots & a_{n-1} & a_n \\ a_1 & a_2 & x & \cdots & a_{n-1} & a_n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & a_2 & a_3 & \cdots & x & a_n \\ a_1 & a_2 & a_3 & \cdots & a_n & x \end{pmatrix}}_{\text{an } (n+1)\times(n+1)\text{-matrix}} = \left(x + \sum_{i=1}^{n} a_i\right) \prod_{i=1}^{n} (x - a_i).$$

Proposition 5.15.5. Let $n \in \mathbb{N}$. Let $x_1, x_2, \ldots, x_n$ be n numbers, and let a be a further number. Then,

$$\det \begin{pmatrix} x_1 & a & a & \cdots & a \\ a & x_2 & a & \cdots & a \\ a & a & x_3 & \cdots & a \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a & a & a & \cdots & x_n \end{pmatrix} = \prod_{i=1}^{n} (x_i - a) + a \sum_{i=1}^{n} y_i,$$

where we set $y_i := \prod_{\substack{k \in \{1,2,\ldots,n\};\\ k \neq i}} (x_k - a)$ for each $i \in \{1, 2, \ldots, n\}$.

Both of these propositions make good exercises in determinant evaluation. (Proposition 5.15.4 is [Grinbe20, Exercise 6.21], while Proposition 5.15.5 is `https://math.stackexch` .)

See [KleSta19] and [Rubey00] for more applications of the Matrix-Tree Theorem, and [Holzer22] for many more related results.

5.16. de Bruijn sequences

5.16.1. Definition

Let me move on to a more intricate application of what we have learned about arborescences.

A little puzzle first: What is special about the periodic sequence

$$||:\ 0000\ 1111\ 0110\ 0101\ :|| \qquad ?$$

(This is an infinite sequence of 0's and 1's; the spaces between some of them are only for readability. The $||:$ and $:||$ symbols are "repeat signs" – they mean that everything that stands between them should be repeated over and over. So the sequence above is 0000 1111 0110 0101 0000 1111 ....)

One nice property of this sequence is that if you slide a ”length-4 window” (i.e., a window that shows four consecutive entries) along it, you get all 16 possible bitstrings of length 4 depending on the position of the window, and these bitstrings do not repeat until you move 16 steps to the right. Just see:

$$\begin{array}{l}
\boxed{0000}11110110010100001111\ldots \\
0\boxed{0001}1110110010100001111\ldots \\
00\boxed{0011}110110010100001111\ldots \\
000\boxed{0111}10110010100001111\ldots \\
0000\boxed{1111}0110010100001111\ldots \\
00001\boxed{1110}110010100001111\ldots \\
000011\boxed{1101}10010100001111\ldots \\
0000111\boxed{1011}0010100001111\ldots \\
00001111\boxed{0110}010100001111\ldots \\
000011110\boxed{1100}10100001111\ldots \\
0000111101\boxed{1001}0100001111\ldots \\
00001111011\boxed{0010}100001111\ldots \\
000011110110\boxed{0101}00001111\ldots \\
0000111101100\boxed{1010}0001111\ldots \\
00001111011001\boxed{0100}001111\ldots \\
000011110110010\boxed{1000}01111\ldots
\end{array}$$

Note that, as you slide the window along the sequence, at each step, the first bit is removed and a new bit is inserted at the end. Thus, by sliding a length-4 window along the above sequence, you run through all 16 possible length-4 bitstrings in such a way that each bitstring is obtained from the previous one by removing the first bit and inserting a new bit at the end. This is nice and somewhat similar to Gray codes (in which you run through all bitstrings of a given length in such a way that only a single bit is changed at each step).

Can we find such nice sequences for any window length, not just 4 ?

Here is an answer for window length 3, for instance:

$$||: \ 000\ 111\ 01 \ :|| \ .$$

What about higher window length?

Moreover, we can ask the same question with other alphabets. For instance, instead of bits, here is a similar sequence for the alphabet $\{0, 1, 2\}$ (that is, we use the numbers $0, 1, 2$ instead of 0 and 1) and window length 2:

$$||: 00\ 11\ 22\ 02\ 1 \ :|| \ .$$

What about the general case? Let us give it a name:

Definition 5.16.1. 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 $q \in \mathbb{Z}$ (so that $c_{k^n} = c_0$ and $c_{k^n+1} = c_1$ and so on).

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 the entries of this 9-tuple as $c_0, c_1, \ldots, c_8$ (and extend the indices periodically, so that $c_9 = c_0$), then we have

$$\begin{aligned} &(0,0) = (c_0, c_1); \qquad &&(0,1) = (c_1, c_2); \qquad &&(0,2) = (c_6, c_7); \\ &(1,0) = (c_8, c_9); &&(1,1) = (c_2, c_3); &&(1,2) = (c_3, c_4); \\ &(2,0) = (c_5, c_6); &&(2,1) = (c_7, c_8); &&(2,2) = (c_4, c_5). \end{aligned}$$

This de Bruijn sequence $(0, 0, 1, 1, 2, 2, 0, 2, 1)$ corresponds to the periodic sequence || : 00 11 22 02 1 : || that we found above.

5.16.2. Existence of de Bruijn sequences

It turns out that de Bruijn sequences always exist:

Theorem 5.16.2 (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 approach 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.

Unfortunately, we don't have any useful criteria that would show that such a cycle exists. So this idea seems to be a dead end.

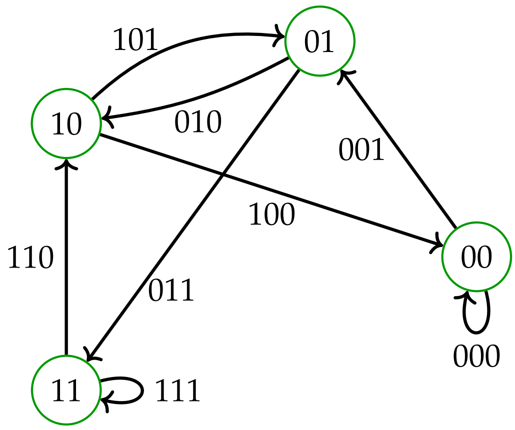
However, let us do something counterintuitive: We try to reinterpret de Bruijn sequences in terms of Eulerian circuits (rather than Hamiltonian cycles), since we have a good criterion for the existence of Eulerian circuits (unlike for that of Hamiltonian cycles)!

We need a different digraph for that. Namely, we let D 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

$$\psi\left(a_1, a_2, \ldots, a_n\right) = \left(\left(a_1, a_2, \ldots, a_{n-1}\right), \ \left(a_2, a_3, \ldots, a_n\right)\right).$$

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 of elements of K, 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 $i \in K^{n-1}$ to each $(n-1)$-tuple $j \in K^{n-1}$ that is obtained by dropping the first entry of i and adding a new entry at the end. (Be careful: If $n = 1$, then D has only one vertex but n arcs. If this confuses you, just do the $n = 1$ case by hand. For any $n > 1$, there are no parallel arcs in D.)

Example 5.16.3. For example, if $n = 3$ and $k = 2$ and $K = \{0, 1\}$, then D looks as follows (we again write our tuples without commas and without parentheses):



Let us make a few observations about D:

- The multidigraph D is strongly connected.

  [*Proof:* We need to show that for any two vertices i and j of D, there is a walk from i to j. But this is easy: Just insert the entries of j into i one by one, pushing out the entries of i. In other words, using the notation k_p for the p-th entry of any tuple k, we have the walk

$$\begin{aligned} i = &(i_1, i_2, \ldots, i_{n-1}) \\ &\to (i_2, i_3, \ldots, i_{n-1}, j_1) \\ &\to (i_3, i_4, \ldots, i_{n-1}, j_1, j_2) \\ &\to \cdots \\ &\to (i_{n-1}, j_1, j_2, \ldots, j_{n-2}) \\ &\to (j_1, j_2, \ldots, j_{n-1}) = j. \end{aligned}$$

Note that this walk has length $n-1$, and is the unique walk from i to j that has length $n-1$. Thus, the # of walks from i to j that have length $n-1$ is 1. This will come useful further below.]

- Thus, the multidigraph D is weakly connected (since any strongly connected digraph is weakly connected).

- The multidigraph D is balanced, and in fact each vertex of D has outdegree k and indegree k.

  [*Proof:* Let i be a vertex of D. The arcs with source i are the n-tuples whose first $n-1$ entries form the $(n-1)$-tuple i while the last, n-th entry is an arbitrary element of K. Thus, there are $|K|$ many such arcs. In other words, i has outdegree k. A similar argument shows that i has indegree k. This entails that $\deg^- i = \deg^+ i$. Since this holds for every vertex i, we conclude that D is balanced.]

- The digraph D has an Eulerian circuit.

  [*Proof:* This follows from the directed Euler–Hierholzer theorem (Theorem 4.7.2), since D is weakly connected and balanced. Alternatively, we can derive this from the BEST theorem (Theorem 5.9.1) as follows: Pick an arbitrary arc a of D, and let r be its source. Then, r is a from-root of D (since D is strongly connected), and thus D has a spanning arborescence rooted from r (by Theorem 5.8.4). In other words, using the notations of the BEST theorem (Theorem 5.9.1), we have $\tau(D, r) \neq 0$. Moreover, each vertex of D has indegree $k > 0$. Thus, the BEST theorem yields

  $$\varepsilon(D, a) = \underbrace{\tau(D, r)}_{\neq 0} \cdot \underbrace{\prod_{u \in V} \left(\deg^- u - 1\right)!}_{\neq 0} \neq 0.$$

  But this shows that D has an Eulerian circuit whose last arc is a.]

So we know that D has an Eulerian circuit $\mathbf{c}$. This Eulerian circuit leads to a de Bruijn sequence as follows:

Let $p_0, p_1, \ldots, p_{k^n-1}$ be the arcs of $\mathbf{c}$ (from first to last). Extend the subscripts periodically modulo k^n (that is, set $p_{q+k^n} = p_q$ for all $q \in \mathbb{N}$). Thus, we obtain an infinite walk[69] with arcs $p_0, p_1, p_2, \ldots$ (since $\mathbf{c}$ is a circuit). In other words, for each $i \in \mathbb{N}$, the target of the arc p_i is the source of the arc p_{i+1}.

In other words, for each $i \in \mathbb{N}$, the last $n-1$ entries of p_i are the first $n-1$ entries of p_{i+1} (since the target of p_i is the tuple consisting of the last $n-1$ entries of p_i, whereas the source of p_{i+1} is the tuple consisting of the first $n-1$ entries of p_{i+1}). Therefore, for each $i \in \mathbb{N}$ and each $j \in \{2, 3, \ldots, n\}$, we have

$$\begin{aligned} & (\text{the } j\text{-th entry of } p_i) \\ & = (\text{the } (j-1)\text{-st entry of } p_{i+1}) . \end{aligned} \tag{25}$$

Now, for each $i \in \mathbb{N}$, we let x_i denote the first entry of the n-tuple p_i. Then, $x_{q+k^n} = x_q$ for all $q \in \mathbb{N}$ (since $p_{q+k^n} = p_q$ for all $q \in \mathbb{N}$). In other words, the sequence $(x_0, x_1, x_2, \ldots)$ repeats itself every k^n terms. Note that the k^n-tuple $(x_0, x_1, \ldots, x_{k^n-1})$ consists of the first entries of the arcs $p_0, p_1, \ldots, p_{k^n-1}$ of $\mathbf{c}$ (by the definition of x_i).

For each $i \in \mathbb{N}$ and each $s \in \{1, 2, \ldots, n\}$, we have

$$\begin{aligned} & (\text{the } s\text{-th entry of } p_i) \\ & = (\text{the } (s-1)\text{-st entry of } p_{i+1}) \qquad (\text{by (25)}) \\ & = (\text{the } (s-2)\text{-nd entry of } p_{i+2}) \qquad (\text{by (25)}) \\ & = (\text{the } (s-3)\text{-rd entry of } p_{i+3}) \qquad (\text{by (25)}) \\ & = \cdots \\ & = (\text{the 1-st entry of } p_{i+s-1}) \\ & = x_{i+s-1} \qquad (\text{since } x_{i+s-1} \text{ was defined as the first entry of } p_{i+s-1}) . \end{aligned}$$

In other words, for each $i \in \mathbb{N}$, the entries of p_i (from first to last) are $x_i, x_{i+1}, \ldots, x_{i+n-1}$. In other words, for each $i \in \mathbb{N}$, we have

$$p_i = (x_i, x_{i+1}, \ldots, x_{i+n-1}) . \tag{26}$$

Now, recall that $\mathbf{c}$ is an Eulerian circuit. Thus, each arc of D appears exactly once among its arcs $p_0, p_1, \ldots, p_{k^n-1}$. In other words, each n-tuple in K^n appears exactly once among $p_0, p_1, \ldots, p_{k^n-1}$ (since the arcs of D are the n-tuples in K^n). In other words, as i ranges from 0 to $k^n - 1$, the n-tuple p_i takes each possible value in K^n exactly once.

In view of (26), we can rewrite this as follows: As i ranges from 0 to $k^n - 1$, the n-tuple $(x_i, x_{i+1}, \ldots, x_{i+n-1})$ takes each possible value in K^n exactly once (since this n-tuple is precisely p_i, as we have shown in the previous paragraph). In other words, for each $(a_1, a_2, \ldots, a_n) \in K^n$, there is a **unique** $r \in \{0, 1, \ldots, k^n - 1\}$ such that $(a_1, a_2, \ldots, a_n) = (x_r, x_{r+1}, \ldots, x_{r+n-1})$.

Hence, the k^n-tuple $(x_0, x_1, \ldots, x_{k^n-1})$ is a de Bruijn sequence of order n on K. This shows that a de Bruijn sequence exists. Theorem 5.16.2 is thus proven.

[69] We have never formally defined infinite walks, but it should be fairly clear what they are.

Example 5.16.4. For $n = 3$ and $k = 2$ and $K = \{0, 1\}$, one possible Eulerian circuit $\mathbf{c}$ of D is

$$(00,\ \mathbf{001},\ 01,\ \mathbf{010},\ 10,\ \mathbf{101},\ 01,\ \mathbf{011},\ 11,\ \mathbf{111},\ 11,\ \mathbf{110},\ 10,\ \mathbf{100},\ 00)$$

(where we have written the arcs in bold for readability). The first entries of the arcs of this circuit form the sequence 0010111, which is indeed a de Bruijn sequence of order 3 on $\{0, 1\}$. Any 3 consecutive entries of this sequence (extended periodically to the infinite sequence $|| : 0010111 : ||$) form the respective arc of $\mathbf{c}$.

□

Theorem 5.16.2 is merely the starting point of a theory. Several specific de Bruijn sequences are known, many of them having peculiar properties. See [Freder82] for a survey of various such sequences[70] (note that they are called "full length nonlinear shift register sequences" in this survey).[71]

There are also several variations on de Bruijn sequences. For some of them, see [ChDiGr92]. (Note that some of the open questions in that paper are still unsolved.) A variation that recently became quite popular is the notion of a "universal cycle for permutations" – a string that contains all "permutations" (more precisely, n-tuples of distinct elements of K) as factors. See [EngVat18] for some recent progress on minimizing the length of such a string, including a contribution by a notorious hacker known as 4chan. (This is no longer really about Eulerian circuits, since some amount of duplication cannot be avoided in these strings.)

5.16.3. Counting de Bruijn sequences

Let us move in a different direction. Having proved the existence of de Bruijn sequences in Theorem 5.16.2, let us try to count them!

Question. Let n and k be two positive integers. Let K be a k-element set. How many de Bruijn sequences of order n on K are there?

To solve this, it makes sense to apply the BEST theorem to the digraph D we have constructed above. Alas, D is not of the form G^{bidir} for some undirected graph G, so we cannot apply the undirected MTT (Matrix-Tree Theorem). However, D is a balanced multidigraph, and for such digraphs, a version of the undirected MTT still holds:

---

[70] Some of these sequences (the "prefer-one" and "prefer-opposite" generators) are just disguised implementations of the algorithm for finding an Eulerian circuit implicit in our proof of the BEST theorem.

[71] My favorite is the one obtained by concatenating all Lyndon words whose length divides n in lexicographically increasing order (assuming that the set K is totally ordered). See [Moreno04] for the details of that construction.

Theorem 5.16.5 (balanced Matrix-Tree Theorem)**.** 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 D. Then:

(a) For any vertex r of D, we have

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \det(L_{\sim r, \sim r}).$$

Moreover, this number does not depend on r.

(b) Let t be an indeterminate. Expand the determinant $\det(tI_n + L)$ (here, I_n denotes 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 $c_0, c_1, \ldots, c_n$ are numbers. (Note that this is the characteristic polynomial of L up to substituting $-t$ for t and multiplying by a power of -1. Some of its coefficients are $c_n = 1$ and $c_{n-1} = \operatorname{Tr} L$ and $c_0 = \det L$.) Then, for any vertex r of D, we have

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \frac{1}{n} c_1.$$

(c) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$. Then, for any vertex r of D, we have

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1}.$$

(d) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of L, listed in such a way that $\lambda_n = 0$. If all vertices of D have outdegree > 0, then

$$(\text{\# of Eulerian circuits of } D) = |A| \cdot \frac{1}{n} \cdot \lambda_1 \lambda_2 \cdots \lambda_{n-1} \cdot \prod_{u \in V} (\deg^+ u - 1)!.$$

(If you identify an Eulerian circuit with its cyclic rotations, then you should drop the $|A|$ factor on the right hand side.)

Proof. **(a)** The equality comes from the MTT (Theorem 5.14.7). It remains to prove that the # of spanning arborescences of D rooted to r does not depend on r. But this is Corollary 5.12.1.

(b) follows from **(a)** as in the undirected graph case (proof of Theorem 5.15.1 **(b)**).[72]

[72] In more detail: Just as we proved in our above proof of Theorem 5.15.1 (for the undirected

(c) follows from **(b)** as in the undirected graph case (proof of Theorem 5.15.1 **(c)**).

(d) Assume that all vertices of D have outdegree > 0. Then,

$$\begin{aligned}&(\text{\# of Eulerian circuits of } D)\\&=\sum_{a\in A}(\text{\# of Eulerian circuits of } D \text{ whose first arc is } a)\,.\end{aligned}$$

However, if $a \in A$ is any arc, and if r is the source of a, then

$$\begin{aligned}&(\text{\# of Eulerian circuits of } D \text{ whose first arc is } a)\\&=(\text{\# of spanning arborescences of } D \text{ rooted to } r)\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!\\&\qquad\qquad(\text{by the BEST' theorem (Theorem 5.10.4)})\\&=\frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!\qquad\qquad(\text{by part } \textbf{(c)})\,.\end{aligned}$$

Hence,

$$\begin{aligned}&(\text{\# of Eulerian circuits of } D)\\&=\sum_{a\in A}\underbrace{(\text{\# of Eulerian circuits of } D \text{ whose first arc is } a)}_{=\frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!}\\&=\sum_{a\in A}\frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!\\&=|A|\cdot\frac{1}{n}\cdot\lambda_1\lambda_2\cdots\lambda_{n-1}\cdot\prod_{u\in V}\left(\deg^+ u-1\right)!.\end{aligned}$$

This proves part **(d)**. □

---

case), we have $c_1=\sum_{r=1}^{n}\det\left(L_{\sim r,\sim r}\right)$. However, part **(a)** shows that the number $\det\left(L_{\sim r,\sim r}\right)$ does not depend on r. Thus, the sum $\sum_{r=1}^{n}\det\left(L_{\sim r,\sim r}\right)$ consists of n equal addends, which can be written as $\det\left(L_{\sim r,\sim r}\right)$ for any vertex r of D. Therefore, this sum can be rewritten as $n\cdot\det\left(L_{\sim r,\sim r}\right)$ for any vertex r of D. Hence, the equality $c_1=\sum_{r=1}^{n}\det\left(L_{\sim r,\sim r}\right)$ can be rewritten as $c_1=n\cdot\det\left(L_{\sim r,\sim r}\right)$ for any vertex r of D. Therefore, $\det\left(L_{\sim r,\sim r}\right)=\frac{1}{n}c_1$ for any vertex r of D. Since part **(a)** yields

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r)=\det\left(L_{\sim r,\sim r}\right),$$

we can rewrite this equality as

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r)=\frac{1}{n}c_1.$$

Now, let's try to solve our question – i.e., let's count the de Bruijn sequences of order n on K.

Recall the digraph D from our above proof of Theorem 5.16.2. We constructed a de Bruijn sequence of order n on K by finding an Eulerian circuit of D. This actually works both ways: The map

$$\begin{aligned} \{\text{Eulerian circuits of } D\} &\to \{\text{de Bruijn sequences of order } n \text{ on } K\}, \\ \mathbf{c} &\mapsto (\text{the sequence of first entries of the arcs of } \mathbf{c}) \end{aligned}$$

is a bijection (make sure you understand why!). Hence, by the bijection principle, we have

$$\begin{aligned} &(\text{\# of de Bruijn sequences of order } n \text{ on } K) \\ &= (\text{\# of Eulerian circuits of } D). \end{aligned} \tag{27}$$

By Theorem 5.16.5 **(d)**, however, we have

$$\begin{aligned} &(\text{\# of Eulerian circuits of } D) \\ &= |K^n| \cdot \frac{1}{k^{n-1}} \cdot \lambda_1 \lambda_2 \cdots \lambda_{k^{n-1}-1} \cdot \prod_{u \in K^{n-1}} \left(\deg^+ u - 1\right)!, \end{aligned} \tag{28}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_{k^{n-1}}$ are the eigenvalues of the Laplacian L of D, indexed in such a way that $\lambda_{k^{n-1}} = 0$. (Note that the digraph $D = \left(K^{n-1}, K^n, \psi\right)$ has k^{n-1} vertices, not n vertices, so the "n" in Theorem 5.16.5 is k^{n-1} here.)

As we know, each vertex of D has outdegree k. That is, we have $\deg^+ u = k$ for each $u \in K^{n-1}$. Thus,

$$\prod_{u \in K^{n-1}} \left(\deg^+ u - 1\right)! = \prod_{u \in K^{n-1}} (k-1)! = ((k-1)!)^{k^{n-1}}.$$

Also,

$$|K^n| \cdot \frac{1}{k^{n-1}} = k^n \cdot \frac{1}{k^{n-1}} = k.$$

It remains to find $\lambda_1 \lambda_2 \cdots \lambda_{k^{n-1}-1}$. What are the eigenvalues of L ?

The Laplacian L of our digraph D is a $k^{n-1} \times k^{n-1}$-matrix whose rows and columns are indexed by $(n-1)$-tuples in K^{n-1}. Strictly speaking, we should relabel the vertices of D as $1, 2, \ldots, k^{n-1}$ here, in order to have a "proper matrix" with a well-defined order on its rows and columns. But let's not do this; instead, I trust you can do the relabeling yourself, or just use the more general notion of matrices that allows for the rows and the columns to be indexed by arbitrary things (see `https://mathoverflow.net/questions/317105` for details).

Let C be the adjacency matrix of the digraph D; this is the $k^{n-1} \times k^{n-1}$-matrix (again with rows and columns indexed by $(n-1)$-tuples in K^{n-1}) whose (i, j)-th entry is the # of arcs with source i and target j. In particular, the trace of C is thus the # of loops of D. It is easy to see that the loops of D are precisely the arcs of the form $(x, x, \ldots, x) \in K^n$ for $x \in K$; thus, D has exactly k loops. Hence, the trace of C is k.

Recall the definition of the Laplacian matrix L. We can restate it as follows:

$$L = \Delta - C, \tag{29}$$

where Δ is the diagonal matrix whose diagonal entries are the outdegrees of the vertices of D. Since each vertex of D has outdegree k, the latter diagonal matrix Δ is simply $k \cdot I$, where I is the identity matrix (of the appropriate size). Hence, (29) can be rewritten as

$$L = k \cdot I - C.$$

Thus, if $\gamma_1, \gamma_2, \ldots, \gamma_{k^{n-1}}$ are the eigenvalues of C, then $k - \gamma_1, k - \gamma_2, \ldots, k - \gamma_{k^{n-1}}$ are the eigenvalues of L. Computing the former will thus help us find the latter.

Furthermore, let J be the $k^{n-1} \times k^{n-1}$-matrix (again with rows and columns indexed by $(n-1)$-tuples in K^{n-1}) whose all entries are 1. It is easy to see that the eigenvalues of J are

$$\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k^{n-1}.$$

(The easiest way to see this is by noticing that J has rank 1 and trace k^{n-1}. [73])

Now, here is something really underhanded: We observe that

$$C^{n-1} = J.$$

[*Proof:* We need to show that all entries of the matrix C^{n-1} are 1. So let i and j be two vertices of D. We must then show that the (i, j)-th entry of C^{n-1} is 1.

Recall the combinatorial interpretation of the powers of an adjacency matrix (Theorem 4.5.10): For any $\ell \in \mathbb{N}$, the (i, j)-th entry of C^ℓ is the # of walks from i to j (in D) that have length ℓ. Thus, in particular, the (i, j)-th entry of C^{n-1} is the # of walks from i to j (in D) that have length $n - 1$. But this number is actually 1, as we have already shown in our above proof of Theorem 5.16.2. This completes the proof of $C^{n-1} = J$.]

How does this help us compute the eigenvalues of C ? Well, let $\gamma_1, \gamma_2, \ldots, \gamma_{k^{n-1}}$ be the eigenvalues of C. Then, for any $\ell \in \mathbb{N}$, the eigenvalues of C^ℓ are $\gamma_1^\ell, \gamma_2^\ell, \ldots, \gamma_{k^{n-1}}^\ell$ (this is a fact that holds for any square matrix, and is probably easiest to prove using the Jordan canonical form or triangularization). Hence, in

[73] Here are the details: The matrix J has rank 1 (since all its rows are the same); thus, all but one of its eigenvalues are 0. It remains to show that the remaining eigenvalue is k^{n-1}. However, it is known that the sums of the eigenvalues of a square matrix equals its trace. Thus, if all but one of the eigenvalues of a square matrix are 0, then the remaining eigenvalue equals its trace. Applying this to our matrix J, we see that its remaining eigenvalue equals its trace, which is k^{n-1}.

particular, $\gamma_1^{n-1}, \gamma_2^{n-1}, \ldots, \gamma_{k^{n-1}}^{n-1}$ are the eigenvalues of $C^{n-1} = J$; but we know that the latter eigenvalues are $\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k^{n-1}$. Hence, all but one of the k^{n-1} numbers $\gamma_1^{n-1}, \gamma_2^{n-1}, \ldots, \gamma_{k^{n-1}}^{n-1}$ equal 0. Thus, all but one of the k^{n-1} numbers $\gamma_1, \gamma_2, \ldots, \gamma_{k^{n-1}}$ equal 0 (we don't know what the remaining number is, since $(n-1)$-st roots are not uniquely determined in $\mathbb{C}$). In other words, all but one of the eigenvalues of C equal 0. The remaining eigenvalue must thus be the trace of C (because the sum of the eigenvalues of a square matrix is known to be the trace of that matrix), and therefore equal k (since we know that the trace of C is k).

So we have shown that the eigenvalues of C are $\underbrace{0, 0, \ldots, 0}_{k^{n-1}-1 \text{ many zeroes}}, k$. Thus, the eigenvalues of L are

$$\underbrace{k-0, \ k-0, \ \ldots, \ k-0}_{k^{n-1}-1 \text{ many } (k-0)\text{'s}}, \ k-k$$

(because if $\gamma_1, \gamma_2, \ldots, \gamma_{k^{n-1}}$ are the eigenvalues of C, then $k-\gamma_1, k-\gamma_2, \ldots, k-\gamma_{k^{n-1}}$ are the eigenvalues of L). In other words, the eigenvalues of L are

$$\underbrace{k, k, \ldots, k}_{k^{n-1}-1 \text{ many } k\text{'s}}, 0.$$

Hence, the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{k^{n-1}-1}$ in (28) all equal k. Thus, (28) simplifies to

$$\begin{aligned}
&(\text{\# of Eulerian circuits of } D) \\
&= \underbrace{|K^n| \cdot \frac{1}{k^{n-1}}}_{= k^n \cdot \frac{1}{k^{n-1}} = k} \cdot \underbrace{kk\cdots k}_{k^{n-1}-1 \text{ factors}} \cdot \underbrace{\prod_{u \in K^{n-1}} \left(\deg^+ u - 1\right)!}_{= ((k-1)!)^{k^{n-1}}} \\
&= \underbrace{k \cdot \underbrace{kk\cdots k}_{k^{n-1}-1 \text{ factors}}}_{= k^{k^{n-1}}} \cdot ((k-1)!)^{k^{n-1}} = k^{k^{n-1}} \cdot ((k-1)!)^{k^{n-1}} \\
&= \Bigg(\underbrace{k \cdot (k-1)!}_{= k!}\Bigg)^{k^{n-1}} = k!^{k^{n-1}}.
\end{aligned}$$

In view of this, we can rewrite (27) as

$$(\text{\# of de Bruijn sequences of order } n \text{ on } K) = k!^{k^{n-1}}.$$

Thus, we have proved the following:

Theorem 5.16.6. Let n and k be positive integers. Let K be a k-element set. Then,

$$(\text{\# of de Bruijn sequences of order } n \text{ on } K) = k!^{k^{n-1}}.$$

What a nice (and huge) answer!

Our above proof of Theorem 5.16.6 is essentially taken from [Stanle18, Chapter 10].

We note that a combinatorial proof of Theorem 5.16.6 (avoiding any use of linear algebra) has been recently given in [BidKis02].

5.17. More on Laplacians

Much more can be said about the Laplacian of a digraph. The study of matrices associated to a graph or digraph is known as **spectral graph theory**; I'd say the Laplacian is probably the most prominent of these matrices (even though the adjacency matrix is somewhat easier to define). The original form of the matrix-tree theorem (actually a subtler variant of Theorem 5.15.1 **(a)**) was found by Gustav Kirchhoff in his study of electricity [Kirchh47] (see [Holzer22, §2.1.1] for a modern exposition); the effective resistance between two nodes of an electrical network is a ratio of spanning-tree counts and thus can be computed using the Laplacian (see, e.g., [Vos16, §2 and §3]). To be more precise, this relies on a "weighted count" of spanning trees, which is more general than the counting we have done so far; we will learn about it in the next section.

Another application of Laplacians is to drawing graphs: see "spectral layout" or "spectral graph drawing" (e.g., [Gallie13]).

5.18. On the left nullspace of the Laplacian

Let me mention one more result about Laplacians of digraphs that answers a rather natural question you might already have asked yourself. Recall that the Laplacian L of a digraph D always satisfies $Le = 0$, where $e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$. Thus, the vector e belongs to the right nullspace (= right kernel) of L. It is not hard to see that if D has a to-root and we are working over a characteristic-0 field, then e spans this nullspace, i.e., there are no vectors in that nullspace other than scalar multiples of e. (This is actually an "if and only if".) What about the left nullspace of L ? Can we explicitly find a nonzero vector f with $fL = 0$? The answer is positive:

Theorem 5.18.1 (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, $fL = 0$.

Theorem 5.18.1 (or, more precisely, its weighted version, which we will see in the next section) can be used to explicitly compute the steady state of a Markov chain (see [KrGrWi10]); a similar interpretation, but in economical terms (emergence of money in a barter economy), appears in [Sahi14, §1].

We shall give a proof of Theorem 5.18.1 based upon two lemmas. The first lemma is a general linear-algebraic result:

Lemma 5.18.2. Let B be an $n \times n$-matrix over an arbitrary commutative ring $\mathbb{K}$. (For example, $\mathbb{K}$ can be $\mathbb{R}$, in which case B is a real matrix.) Assume that the sum of all columns of B is the zero vector. Then, for any $r, s, t \in \{1, 2, \ldots, n\}$, we have

$$\det(B_{\sim r, \sim t}) = (-1)^{s-t} \det(B_{\sim r, \sim s}).$$

Proof of Lemma 5.18.2. There are various ways to prove this, but here is probably the most elegant one:

We WLOG assume that $s \neq t$, since otherwise the claim is obvious. Let us now change the r-th row of the matrix B as follows:

- We replace the s-th entry of the r-th row by 1.
- We replace the t-th entry of the r-th row by -1.
- We replace all other entries of the r-th row by 0.

Let C be the resulting $n \times n$-matrix.[74] Thus, C agrees with B in all rows other than the r-th one. Hence, in particular,

$$C_{\sim r, \sim k} = B_{\sim r, \sim k} \qquad \text{for each } k \in \{1, 2, \ldots, n\}. \tag{30}$$

[74]For example, if $n = 4$ and $B = \begin{pmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \\ a''' & b''' & c''' & d''' \end{pmatrix}$ and $s = 1$ and $t = 3$ and $r = 2$, then $C = \begin{pmatrix} a & b & c & d \\ 1 & 0 & -1 & 0 \\ a'' & b'' & c'' & d'' \\ a''' & b''' & c''' & d''' \end{pmatrix}$.

Note also that the only nonzero entries in the r-th row of C are[75] $C_{r,s} = 1$ and $C_{r,t} = -1$. Hence, the entries in the r-th row of C add up to 0.

Recall that the sum of all columns of B is the zero vector. In other words, in each row of B, the entries add up to 0. The matrix C therefore also has this property (because the only row of C that differs from the corresponding row of B is the r-th row; however, we have shown above that in the r-th row, the entries of C also add up to 0). In other words, the sum of all columns of C is the zero vector. This easily entails that $\det C = 0$ [76].

On the other hand, Laplace expansion along the r-th row yields

$$\begin{aligned}\det C &= \sum_{k=1}^{n} (-1)^{r+k} C_{r,k} \det (C_{\sim r,\sim k}) \\ &= (-1)^{r+s} 1 \det (C_{\sim r,\sim s}) + (-1)^{r+t} (-1) \det (C_{\sim r,\sim t})\end{aligned}$$

(since the only nonzero entries $C_{r,k}$ in the r-th row of C are $C_{r,s} = 1$ and $C_{r,t} = -1$). Comparing this with $\det C = 0$, we obtain

$$\begin{aligned}0 &= (-1)^{r+s} 1 \det (C_{\sim r,\sim s}) + (-1)^{r+t} (-1) \det (C_{\sim r,\sim t}) \\ &= (-1)^{r+s} \det \underbrace{(C_{\sim r,\sim s})}_{\substack{=B_{\sim r,\sim s} \\ \text{(by (30))}}} - (-1)^{r+t} \det \underbrace{(C_{\sim r,\sim t})}_{\substack{=B_{\sim r,\sim t} \\ \text{(by (30))}}} \\ &= (-1)^{r+s} \det (B_{\sim r,\sim s}) - (-1)^{r+t} \det (B_{\sim r,\sim t}) .\end{aligned}$$

In other words, $(-1)^{r+t} \det (B_{\sim r,\sim t}) = (-1)^{r+s} \det (B_{\sim r,\sim s})$. Dividing both sides of this by $(-1)^{r+t}$, we obtain $\det (B_{\sim r,\sim t}) = (-1)^{s-t} \det (B_{\sim r,\sim s})$. This proves Lemma 5.18.2. ☐

Our next lemma is the following generalization of Theorem 5.14.7:

Theorem 5.18.3 (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 two vertices of D. Then,

$$(\text{\# of spanning arborescences of } D \text{ rooted to } r) = (-1)^{r+s} \det (L_{\sim r,\sim s}) .$$

[75] We are using the notation $C_{r,k}$ for the entry of C in the r-th row and the k-th column.

[76] *Proof.* It is well-known that the determinant of a matrix does not change if we add a column to another. Hence, the determinant of C will not change if we add each column of C other than the first one to the first column of C. However, the result of this operation will be a matrix whose first column is 0 (since the sum of all columns of C is the zero vector), and therefore this matrix will have determinant 0. Since the operation did not change the determinant, we thus conclude that the determinant of C was 0. In other words, $\det C = 0$.

Note that Theorem 5.14.7 is the particular case of Theorem 5.18.3 for $s = r$. Fortunately, using Lemma 5.18.2, we can easily derive the general case from the particular:

Proof of Theorem 5.18.3. We have seen (in the proof of Proposition 5.14.6) that the sum of all columns of the Laplacian L is the zero vector. Hence, Lemma 5.18.2 (applied to $\mathbb{K} = \mathbb{Q}$ and $B = L$ and $t = r$) yields

$$\det\left(L_{\sim r,\sim r}\right) = \underbrace{(-1)^{s-r}}_{=(-1)^{r+s}} \det\left(L_{\sim r,\sim s}\right) = (-1)^{r+s} \det\left(L_{\sim r,\sim s}\right).$$

However, the Matrix-Tree Theorem (Theorem 5.14.7) yields

$$\begin{aligned}(\text{\# of spanning arborescences of } D \text{ rooted to } r) &= \det\left(L_{\sim r,\sim r}\right) \\ &= (-1)^{r+s} \det\left(L_{\sim r,\sim s}\right).\end{aligned}$$

This proves Theorem 5.18.3. ☐

We are now ready to prove Theorem 5.18.1:

Proof of Theorem 5.18.1. For each $r, s \in \{1, 2, \ldots, n\}$, we have

$$\begin{aligned}\tau(D, r) &= (\text{\# of spanning arborescences of } D \text{ rooted to } r) \\ &\qquad (\text{by the definition of } \tau(D, r)) \\ &= (-1)^{r+s} \det\left(L_{\sim r,\sim s}\right) \qquad (31)\end{aligned}$$

(by Theorem 5.18.3).

However, we have $f = (\tau(D,1), \tau(D,2), \ldots, \tau(D,n))$. Thus, for each $s \in \{1, 2, \ldots, n\}$, the s-th entry of the column vector fL is[77]

$$\begin{aligned}&\sum_{r=1}^{n} \underbrace{\tau(D, r)}_{\substack{=(-1)^{r+s}\det(L_{\sim r,\sim s}) \\ \text{(by (31))}}} L_{r,s} \\ &= \sum_{r=1}^{n} (-1)^{r+s} \det\left(L_{\sim r,\sim s}\right) L_{r,s} \\ &= \sum_{r=1}^{n} (-1)^{r+s} L_{r,s} \det\left(L_{\sim r,\sim s}\right) = \det L \\ &\qquad \left(\begin{array}{c}\text{since Laplace expansion along the } s\text{-th column} \\ \text{yields } \det L = \sum\limits_{r=1}^{n} (-1)^{r+s} L_{r,s} \det\left(L_{\sim r,\sim s}\right)\end{array}\right) \\ &= 0\end{aligned}$$

(by Proposition 5.14.6). This shows that all entries of fL are 0. In other words, $fL = 0$. Theorem 5.18.1 is thus proved. ☐

[77] We are using the notation $L_{r,s}$ for the entry of the matrix L in the r-th row and the s-th column.

Other proofs of Theorem 5.18.1 exist. In particular, a combinatorial proof is sketched in [Sahi14, Theorem 1]. (More precisely, [Sahi14, Theorem 1] in this paper is the claim of Theorem 5.18.1 upon reversing all the arcs and replacing all matrices by their transposes.)[78]

5.19. A weighted Matrix-Tree Theorem

5.19.1. 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 object you are interested in), and then you **sum** the weights of all arborescences (instead of merely counting them). This generalizes counting, because if all weights are 1, then you get the # of arborescences.

If you pick the weights to be completely random, then the sum won't usually be particularly interesting. However, some choices of weights lead to good behavior. Let us see what we get if we assign a weight to each **arc** of our digraph, and then define the weight of an arborescence to be the **product** of the weights of the arcs that appear in this arborescence.

Definition 5.19.1. 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. (You can assume that $\mathbb{K} = \mathbb{R}$, so that the weights are just numbers.)

(a) For any two vertices $i, j \in V$, we let $a^w_{i,j}$ be the sum of the weights of all arcs of D that have source i and target j.

(b) For any vertex $i \in V$, we define the **weighted outdegree** $\deg^{+w} i$ of i to be the sum
$$\sum_{\substack{a \in A;\\ \text{the source of } a \text{ is } i}} w_a.$$

(c) If B is a subdigraph of D, then the **weight** $w(B)$ of B is defined to be the product $\prod_{a \text{ is an arc of } B} w_a$. This is the product 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 $L^w \in \mathbb{K}^{n \times n}$ (note that the "w" here is a superscript, not an exponent) whose entries are given by
$$L^w_{i,j} = \left(\deg^{+w} i\right) \cdot [i = j] - a^w_{i,j} \qquad \text{for all } i, j \in V.$$

[78] I tried to explain this proof in more detail in the solutions to Spring 2018 Math 4707 midterm #3 – see the proof of Theorem 0.7 in those solutions; you be the judge if I succeeded.

These definitions generalize analogous definitions in the "unweighted case". Indeed, if we take all the arc weights w_a to be 1, then the weighted outdegree $\deg^{+w} i$ of a vertex i becomes its usual outdegree $\deg i$, and the weighted Laplacian L^w becomes the usual Laplacian L. The weight $w(B)$ of a subdigraph B simply becomes 1 in this case.

5.19.2. The weighted Matrix-Tree Theorem

We now can generalize the original MTT (= Matrix-Tree Theorem)[79] as follows:

Theorem 5.19.2 (weighted Matrix-Tree Theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph.

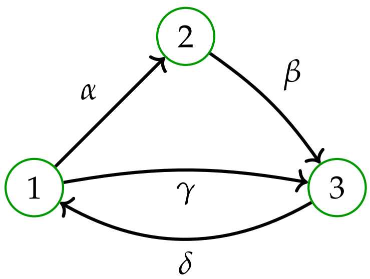
Let $\mathbb{K}$ be a commutative ring. Assume that an element $w_a \in \mathbb{K}$ is assigned to each arc $a \in A$. We call this w_a the **weight** of the arc a.

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,

$$\sum_{\substack{B \text{ is a spanning} \\ \text{arborescence} \\ \text{of } D \text{ rooted to } r}} w(B) = \det\left(L^w_{\sim r, \sim r}\right).$$

Example 5.19.3. Let D be the following multidigraph:

$D =$ 

, and let $r = 3$.

Then, D has two spanning arborescences rooted to r. One of the two has arcs α and β (and thus has weight $w_\alpha w_\beta$); the other has arcs γ and β (and thus has weight $w_\gamma w_\beta$). Hence,

$$\sum_{\substack{B \text{ is a spanning} \\ \text{arborescence} \\ \text{of } D \text{ rooted to } r}} w(B) = w_\alpha w_\beta + w_\gamma w_\beta, \tag{32}$$

The weighted Laplacian L^w is

$$L^w = \begin{pmatrix} w_\alpha + w_\gamma & -w_\alpha & -w_\gamma \\ 0 & w_\beta & -w_\beta \\ -w_\delta & 0 & w_\delta \end{pmatrix}$$

[79] To remind: The original MTT is Theorem 5.14.7.

(since, for example, $\deg^{+w} 1 = w_\alpha + w_\gamma$ and $a^w_{1,1} = 0$ and $a^w_{1,2} = w_\alpha$). Thus,

$$L^w_{\sim 3, \sim 3} = \begin{pmatrix} w_\alpha + w_\gamma & -w_\alpha \\ 0 & w_\beta \end{pmatrix} \qquad \text{and therefore}$$
$$\det \left(L^w_{\sim 3, \sim 3} \right) = \left(w_\alpha + w_\gamma \right) w_\beta = w_\alpha w_\beta + w_\gamma w_\beta.$$

The right hand side of this agrees with that of (32). This confirms the weighted MTT for our D and r.

As we already said, the weighted MTT generalizes the original MTT, because if we take all w_a's to be 1, we just recover the original MTT.

However, we can also go backwards: we can derive the weighted MTT from the original MTT. Let us do this.

5.19.3. The polynomial identity trick

First, we recall a standard result in algebra, known as the **principle of permanence of polynomial identities** or as the **polynomial identity trick** (it also goes under several other names). Here is one incarnation of this principle:

Theorem 5.19.4 (principle of permanence of polynomial identities)**.** 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

$$P(k_1, k_2, \ldots, k_m) = Q(k_1, k_2, \ldots, k_m) \qquad (33)$$

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 identical as polynomials (so that, in particular, the equality (33) holds not only for every $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$, but also for every $(k_1, k_2, \ldots, k_m) \in \mathbb{C}^m$, and more generally, for every $(k_1, k_2, \ldots, k_m) \in \mathbb{K}^m$ where $\mathbb{K}$ is an arbitrary commutative ring).

Theorem 5.19.4 is often summarized as "in order to prove that two polynomials are equal, it suffices to show that they are equal on all nonnegative integer points" (where a "nonnegative integer point" means a point – i.e., a tuple of inputs – whose all entries are nonnegative integers). Even shorter, one says that "a polynomial identity (i.e., an equality between two polynomials) needs only to be checked on nonnegative integers". For example, if you can prove the equality

$$(x+y)^4 + (x-y)^4 = 2x^4 + 12x^2y^2 + 2y^4$$

for all nonnegative integers x and y, then you automatically conclude that this equality holds as a polynomial identity, and thus is true for any elements x and y of a commutative ring.

A typical application of Theorem 5.19.4 is to argue that a polynomial identity you have proved for all nonnegative integers must automatically hold for all inputs (because of Theorem 5.19.4). Some examples of such reasoning can be found in [19fco, §2.6.3 and §2.6.4]. A variant of Theorem 5.19.4 is [Conrad21, Theorem 2.6]; actually, the proof of [Conrad21, Theorem 2.6] can be trivially adapted to prove Theorem 5.19.4 (just replace "nonempty open set in $\mathbb{C}^k$" by "$\mathbb{N}^k$"). In truth, there is nothing special about nonnegative integers and the set $\mathbb{N}$; you could replace $\mathbb{N}$ by any infinite set of numbers (or even any sufficiently large set of numbers, where "sufficiently large" means "more than $\max\{\deg P, \deg Q\}$ many"). See [Alon02, Lemma 2.1] for a fairly general version of Theorem 5.19.4 that includes such cases[80].

5.19.4. Proof of the weighted MTT

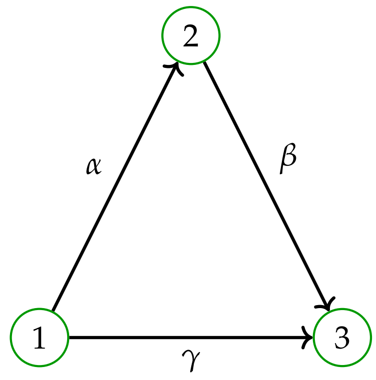
We can now deduce the weighted MTT from the original MTT (Theorem 5.14.7):

Proof of Theorem 5.19.2. The claim of Theorem 5.19.2 (for fixed D and r) is an equality between two polynomials in the arc weights w_a. (For instance, in Example 5.19.3, this equality is $w_\alpha w_\beta + w_\gamma w_\beta = \det \begin{pmatrix} w_\alpha + w_\gamma & -w_\alpha \\ 0 & w_\beta \end{pmatrix}$.)

Therefore, thanks to Theorem 5.19.4, it suffices to prove this equality in the case when all arc weights w_a are nonnegative integers. So let us WLOG assume that arc weights w_a are nonnegative integers.

Let us now replace each arc a of D by w_a many copies of the arc a (having the same source as a and the same target as a). The result is a new digraph D'. Here is an example:

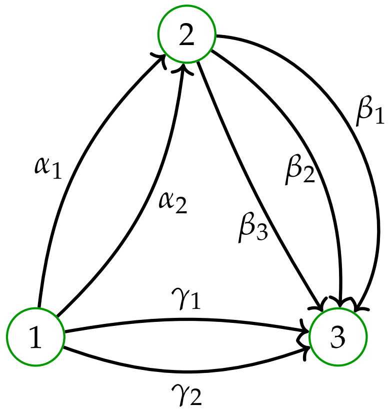
Example 5.19.5. Let D be the digraph

$D =$ 

,

[80] To be precise, [Alon02, Lemma 2.1] is not concerned with two polynomials being identical, but rather with one polynomial being identically zero. But this is an equivalent question: Two polynomials P and Q are identical if and only if their difference $P - Q$ is identically zero.

and let the arc weights be $w_\alpha = 2$ and $w_\beta = 3$ and $w_\gamma = 2$. Then, D' looks as follows:

$D' =$ 

,

where α_1, α_2 are the two arcs obtained from α, and so on.

Now, recall that the digraph D' has the same vertices as D, but each arc a of D has turned into w_a arcs of D'. Thus, the weighted outdegree $\deg^{+w} i$ of a vertex i of D equals the (usual, i.e., non-weighted) outdegree $\deg^+ i$ of the same vertex i of D'. Hence, the weighted Laplacian L^w of D is the (usual, i.e., non-weighted) Laplacian of D'.

Recall again that the digraph D' has the same vertices as D, but each arc a of D has turned into w_a arcs of D'. Thus, each subdigraph B of D gives rise to $w(B)$ many subdigraphs of D' (because we can replace each arc a of B by any of the w_a many copies of this arc in D'). Moreover, this correspondence takes spanning arborescences to spanning arborescences[81], and we can obtain any spanning arborescence of D' in this way from exactly one B. Hence,

$$\sum_{\substack{B \text{ is a spanning} \\ \text{arborescence} \\ \text{of } D \text{ rooted to } r}} w(B) = (\text{\# of spanning arborescences of } D' \text{ rooted to } r).$$

Thus, applying the original MTT (Theorem 5.14.7) to D' yields the weighted MTT for D (since the weighted Laplacian L^w of D is the (usual, i.e., non-weighted) Laplacian of D'). This completes the proof of Theorem 5.19.2.

[*Remark:* Alternatively, it is not hard to adapt our above proof of the original MTT to the weighted case.] □

5.19.5. Application: Counting trees by their degrees

The weighted MTT has some applications that wouldn't be obvious from the original MTT. Here is one:

[81] More precisely: Let B be a subdigraph of D, and let B' be any of the $w(B)$ many subdigraphs of D' that are obtained from B through this correspondence. Then, B is a spanning arborescence of D rooted to r if and only if B' is a spanning arborescence of D' rooted to r.

Exercise 5.27. 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 from Corollary 5.14.9 that there are n^{n-2} many n-trees. How many of these n-trees have the property that

$$\deg i = d_i \qquad \text{for each vertex } i \text{ ?}$$

Solution. The n-trees are just the spanning trees of the complete graph K_n.

To incorporate the $\deg i = d_i$ condition into our count, we use a generating function. So let us **not** fix the numbers $d_1, d_2, \ldots, d_n$, but rather consider the polynomial

$$P(x_1, x_2, \ldots, x_n) := \sum_{T \text{ is a } n\text{-tree}} x_1^{\deg 1} x_2^{\deg 2} \cdots x_n^{\deg n} \tag{34}$$

in n indeterminates $x_1, x_2, \ldots, x_n$ (where $\deg i$ means the degree of i in T). 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

$$\deg i = d_i \qquad \text{for each vertex } i$$

(because each such n-tree T contributes a monomial $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$ to the sum on the right hand side of (34), whereas any other n-tree T contributes a different monomial to this sum).

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 follows:

$$P(x_1, x_2, \ldots, x_n) = \sum_{T \text{ is an } n\text{-tree}} w(T),$$

where $w(T)$ denotes the product of the weights of all edges of T. (Indeed, for any subgraph T of K_n, the weight $w(T)$ equals $x_1^{\deg 1} x_2^{\deg 2} \cdots x_n^{\deg n}$, where $\deg i$ means the degree of i in T.)

We have assigned weights to the edges of the graph K_n; let us now assign the same weights to the arcs 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

$$w_{(ij,1)} = w_{(ij,2)} = w_{ij} = x_i x_j. \tag{35}$$

As we are already used to, we can replace spanning trees of K_n by spanning arborescences of K_n^{bidir} rooted to 1, since the former are in bijection with the latter. Thus, we have

$$\begin{aligned} &(\text{\# of spanning trees of } K_n) \\ &= \left(\text{\# of spanning arborescences of } K_n^{\text{bidir}} \text{ rooted to } 1\right). \end{aligned}$$

Moreover, since this bijection preserves weights (because of (35)), we also have

$$\sum_{\substack{T \text{ is a spanning} \\ \text{tree of } K_n}} w(T) = \sum_{\substack{B \text{ is a spanning} \\ \text{arborescence of } K_n^{\text{bidir}} \\ \text{rooted to } 1}} w(B).$$

In other words,

$$\sum_{T \text{ is an } n\text{-tree}} w(T) = \sum_{\substack{B \text{ is a spanning} \\ \text{arborescence of } K_n^{\text{bidir}} \\ \text{rooted to } 1}} w(B)$$

(since the spanning trees of K_n are precisely the n-trees).

To compute the right hand side, we shall use the weighted Matrix-Tree Theorem. The weighted Laplacian of K_n^{bidir} (with the weights we have just defined) is the $n \times n$-matrix L^w with entries given by

$$\begin{aligned}
L^w_{i,j} &= \left(\deg^{+w} i\right) \cdot [i = j] - a^w_{i,j} \\
&= \begin{cases} \deg^{+w} i - a^w_{i,j}, & \text{if } i = j; \\ -a^w_{i,j}, & \text{if } i \neq j \end{cases} \\
&= \begin{cases} \deg^{+w} i, & \text{if } i = j; \\ -a^w_{i,j}, & \text{if } i \neq j \end{cases} \qquad \left(\begin{array}{c} \text{since } a^w_{i,j} = 0 \text{ when } i = j \\ \text{(because } K_n^{\text{bidir}} \text{ has no loops)} \end{array}\right) \\
&= \begin{cases} x_i(x_1 + x_2 + \cdots + x_n) - x_i x_j, & \text{if } i = j; \\ -x_i x_j, & \text{if } i \neq j \end{cases} \\
&\qquad \left(\begin{array}{c} \text{since } \deg^{+w} i = x_i x_1 + x_i x_2 + \cdots + x_i x_{i-1} + x_i x_{i+1} + \cdots + x_i x_n \\ = x_i(x_1 + x_2 + \cdots + x_{i-1} + x_{i+1} + \cdots + x_n) \\ = x_i(x_1 + x_2 + \cdots + x_n) - x_i x_i \\ = x_i(x_1 + x_2 + \cdots + x_n) - x_i x_j \text{ whenever } i = j, \\ \text{and since } a^w_{i,j} = x_i x_j \text{ whenever } i \neq j \end{array}\right) \\
&= [i = j]\, x_i(x_1 + x_2 + \cdots + x_n) - x_i x_j \\
&= x_i\left([i = j](x_1 + x_2 + \cdots + x_n) - x_j\right).
\end{aligned}$$

We can find its minor $\det\left(L^w_{\sim 1, \sim 1}\right)$ without too much trouble (e.g., using row transformations similar to the ones we have done back in the proof of Cayley's formula[82]); the result is

$$\det\left(L^w_{\sim 1, \sim 1}\right) = x_1 x_2 \cdots x_n (x_1 + x_2 + \cdots + x_n)^{n-2}.$$

[82]The first step, of course, is to factor an x_i out of the i-th row for each i.

Summarizing what we have done so far,

$$\begin{aligned}
P(x_1, x_2, \ldots, x_n) &= \sum_{T \text{ is an } n\text{-tree}} w(T) = \sum_{\substack{B \text{ is a spanning} \\ \text{arborescence of } K_n^{\text{bidir}} \\ \text{rooted to } 1}} w(B) \\
&= \det\left(L^w_{\sim 1, \sim 1}\right) \qquad \text{(by the weighted Matrix-Tree Theorem)} \\
&= x_1 x_2 \cdots x_n (x_1 + x_2 + \cdots + x_n)^{n-2}. \qquad (36)
\end{aligned}$$

As we recall, we are looking for the $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$-coefficient in this polynomial. From (36), we see that

$$\begin{aligned}
&\left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient of } P(x_1, x_2, \ldots, x_n)\right) \\
&= \left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient of } x_1 x_2 \cdots x_n (x_1 + x_2 + \cdots + x_n)^{n-2}\right) \\
&= \left(\text{the } x_1^{d_1-1} x_2^{d_2-1} \cdots x_n^{d_n-1}\text{-coefficient of } (x_1 + x_2 + \cdots + x_n)^{n-2}\right)
\end{aligned}$$

(because when we multiply a polynomial by $x_1 x_2 \cdots x_n$, all the exponents in it get incremented by 1, so its coefficients just shift by a 1 in each exponent).

Now, how can we describe the coefficients of $(x_1 + x_2 + \cdots + x_n)^{n-2}$, or, more generally, of $(x_1 + x_2 + \cdots + x_n)^m$ for some $m \in \mathbb{N}$? These are the so-called **multinomial coefficients** (named in analogy to the binomial coefficients, which are their particular case for $n = 2$). Their definition is as follows: If $p_1, p_2, \ldots, p_n, q$ are nonnegative integers with $q = p_1 + p_2 + \cdots + p_n$, then the **multinomial coefficient** $\dbinom{q}{p_1, p_2, \ldots, p_n}$ is defined to be $\dfrac{q!}{p_1! p_2! \cdots p_n!}$. If $q \neq p_1 + p_2 + \cdots + p_n$, then it is defined to be 0 instead. In either case, this coefficient is easily seen to be an integer.[83] The **multinomial formula (aka multinomial theorem)** says that for each $k \in \mathbb{N}$, we have

$$\begin{aligned}
(x_1 + x_2 + \cdots + x_n)^k &= \sum_{\substack{i_1, i_2, \ldots, i_n \in \mathbb{N}; \\ i_1 + i_2 + \cdots + i_n = k}} \binom{k}{i_1, i_2, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \\
&= \sum_{i_1, i_2, \ldots, i_n \in \mathbb{N}} \binom{k}{i_1, i_2, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}
\end{aligned}$$

(it does not matter whether we restrict the sum by the condition $i_1 + i_2 + \cdots + i_n = k$ or not, since the coefficient $\dbinom{k}{i_1, i_2, \ldots, i_n}$ is defined to be 0 when this condition is violated anyway). Hence,

$$\left(\text{the } x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}\text{-coefficient of } (x_1 + x_2 + \cdots + x_n)^k\right) = \binom{k}{i_1, i_2, \ldots, i_n}$$

[83] See [23wd, Lecture 18, Section 4.12] for an introduction to multinomial coefficients.

for any $k \in \mathbb{N}$ and any $i_1, i_2, \ldots, i_n \in \mathbb{N}$. In particular,

$$\begin{aligned}&\left(\text{the } x_1^{d_1-1} x_2^{d_2-1} \cdots x_n^{d_n-1}\text{-coefficient of } (x_1+x_2+\cdots+x_n)^{n-2}\right) \\ &= \binom{n-2}{d_1-1,\ d_2-1,\ \ldots,\ d_n-1}.\end{aligned}$$

Summarizing, we find

$$\begin{aligned}&\left(\text{the } x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}\text{-coefficient of } P(x_1, x_2, \ldots, x_n)\right) \\ &= \left(\text{the } x_1^{d_1-1} x_2^{d_2-1} \cdots x_n^{d_n-1}\text{-coefficient of } (x_1+x_2+\cdots+x_n)^{n-2}\right) \\ &= \binom{n-2}{d_1-1,\ d_2-1,\ \ldots,\ d_n-1}.\end{aligned}$$

However, the $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$-coefficient of $P(x_1, x_2, \ldots, x_n)$ is the # of n-trees T satisfying the property that

$$\deg i = d_i \qquad \text{for each vertex } i$$

(as we have seen above). Thus, we have proved the following: □

Theorem 5.19.6 (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

$$\binom{n-2}{d_1-1,\ d_2-1,\ \ldots,\ d_n-1}.$$

5.19.6. The weighted harmonic vector theorem

The harmonic vector theorem for Laplacians (Theorem 5.18.1) also has a weighted version:

Theorem 5.19.7 (harmonic vector theorem for weighted Laplacians)**.** Let $D = (V, A, \psi)$ be a multidigraph, where $V = \{1, 2, \ldots, n\}$ for some $n \in \mathbb{N}$. Let $\mathbb{K}$ be a commutative ring. Assume that an element $w_a \in \mathbb{K}$ is assigned to each arc $a \in A$. For each $r \in V$, let $\tau^w(D, r)$ be the sum of the weights of all the spanning arborescences of D rooted to r. Let f^w be the row vector $(\tau^w(D,1),\ \tau^w(D,2),\ \ldots,\ \tau^w(D,n))$. Let L^w be the weighted Laplacian of D. Then, $f^w L^w = 0$.

Proof. Similar to the unweighted case. □

Here ends our study of spanning trees and their enumeration. An interested reader can learn more from [Rubey00], [Holzer22], [Moon70] and [GrSaSu14].