Discrete Structures

Fabian Bosshard

July 13, 2025

Contents

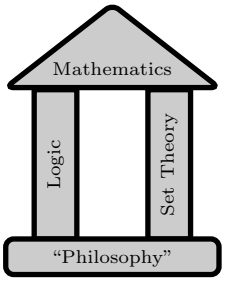
Preface

This document is an unofficial student-made summary of the course Discrete Structures taught by Stefan Wolf in Spring 2025 at the Università della Svizzera italiana. It is mainly based on [1]. It is not complete (e.g. Chapter 1 and 6 from [1] are missing) and could contain errors. If you spot one, please report it to fabianlucasbosshard@gmail.com or open an issue at `https://github.com/fabianbosshard/USI_summaries`.

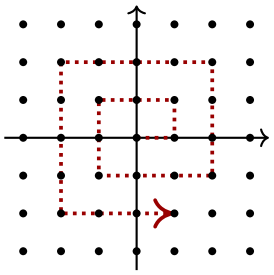
References

[1] Cecilia Boschini, Arne Hansen, and Stefan Wolf. Discrete Mathematics. vdf Hochschulverlag AG an der ETH Zürich, 2022. ISBN: 9783728141101. DOI: 10.3218/4110-1. URL: https://vdf.ch/discrete-mathematics-e-book.html.

1 Introduction



Pillars of mathematics


Matching $\mathbb{Q}$ and $\mathbb{N}$

One can single out two pillars of today's mathematics. Logic determines how to reason within mathematics, i.e., what is considered a valid proof. Set theory describes the objects we deal with.

Definition 1 (Countable set)**.** A set S is called countable if there exists a injective function $f : S \to \mathbb{N}$. If f is also surjective, then S is called countably infinite. ✍

The rational numbers are countable, as we can see if we arrange them in a grid:

$$\mathbb{Q} := \left\{ \frac{a}{b} \in \mathbb{R} \mid a \in \mathbb{Z}, b \in \mathbb{Z} \setminus \{0\} \right\}$$

Theorem 1 (Cantor's diagonal argument)**.** The set of real numbers in the interval $[0, 1]$ is uncountable. Thus the size of the set of real numbers is strictly greater than the size of the set of natural numbers (in the sense that there is no bijective map between the two):

$$|[0, 1]| = |\mathbb{R}| > |\mathbb{N}| = |\mathbb{Q}|. \qquad \triangleleft$$

Proof. (Contradiction) Assume there exists a bijection

$$f : \mathbb{N} \to [0, 1].$$

Then every real number in $[0, 1]$ can be expressed in its binary expansion, allowing us to write:

$$\begin{array}{rcl} f(1) & = & 0.b_{11}b_{12}b_{13}b_{14}\ldots \\ f(2) & = & 0.b_{21}b_{22}b_{23}b_{24}\ldots \\ f(3) & = & 0.b_{31}b_{32}b_{33}b_{34}\ldots \\ f(4) & = & 0.b_{41}b_{42}b_{43}b_{44}\ldots \\ \vdots & & \ddots \end{array}$$

where each b_{ij} is either 0 or 1.

Using Cantor's diagonal method, we now construct a new number x in $[0, 1]$ by flipping the i-th bit of the binary expansion of $f(i)$. Define x as

$$x = 0.c_1c_2c_3c_4\ldots \quad \text{with} \quad c_i = \begin{cases} 1, & \text{if } b_{ii} = 0 \\ 0, & \text{if } b_{ii} = 1 \end{cases}.$$

This guarantees that for every $i \in \mathbb{N}$, the i-th digit of x is different from the i-th digit of $f(i)$.

Since x differs from each $f(i)$ in at least the i-th digit, it follows that x cannot equal any of the numbers in the list generated by f. This contradicts the original assumption that f is a bijection since x is a member of $[0, 1]$ that is not included in the enumeration.

Thus, there is no bijection between $\mathbb{N}$ and $[0, 1]$, which implies that $[0, 1]$ (and hence $\mathbb{R}$) is uncountable. □

2 Propositional Logic

Definition 2 (Proposition, Atom, Connective)**.** A proposition is a sentence, expression, or formula which is either true or false, i.e., truth-definite. An atom or atomic proposition is a basic proposition that is not composed of other propositions. A connective links (generally) two propositions to a new proposition. ✍

Connectives are full characterized by a truth table. As truth can take one of two values, just like bits in a computer, there is a close relation between connectives and logical gates.

2.1 Syntax of propositional logic

In syntax (of logic as well as of programming languages), we specify what a correct string (formula, program) is, independently of its respective "meaning" of "function".

Definition 3 (Syntax)**.** Syntactically correct formulas $\mathcal{E}_\mathcal{D}$ with a set of atoms $\mathcal{D} := \{A, B, C, \dots\}$ are

- atomic formulas in $\mathcal{D}$;
- if f and g are syntactically correct formulas, then also $(\neg f)$, $(f \wedge g)$ and $(f \vee g)$ are syntactically correct.

These are all the syntactically correct formulas. ✍

2.2 Semantics of propositional logic

Definition 4 (Assignment)**.** A truth assignment is a mapping from the set of atoms $\mathcal{D}$ to the set of truth values $\{0, 1\}$. The set of all truth assignments is denoted by $\mathcal{A}$. A truth assignment $\mathcal{A}\colon \mathcal{D} \to \{0, 1\}$ is a function that assigns a truth value to every atom. This truth function is extended to all syntactically correct formulas by simply evaluating the formula, based on the truth values for its atoms and the connectives used. ✍

Definition 5 (Semantic Behavior / Truth Vector)**.** The semantic behavior or truth vector of a proposition is the list of truth values for all possible assignments. ✍

This behavior or vector can be expressed completely in a truth table.

Definition 6 (Semantic Equivalence)**.** Two formulas are semantically equivalent if they have the same truth value for all assigments of their atomic formulas. We write $F \equiv G$ or $F \Leftrightarrow G$. ✍

We introduce the "truth values" 0 and 1 as syntactically correct formulas, abbreviating (the shortest) unsatisfiable formula and tautology, respectively:

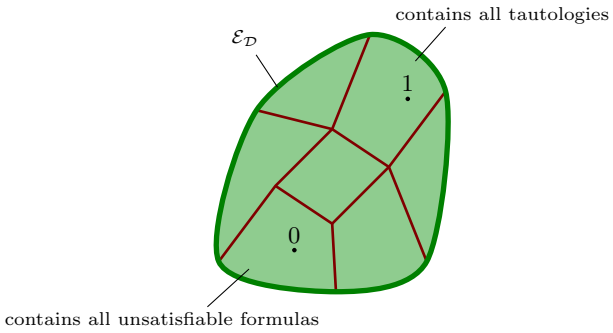
$$0 :\equiv (A \wedge (\neg A))$$
$$1 :\equiv (A \vee (\neg A))$$

Definition 7 (Tautology, Unsatisfiable Formula)**.** If a formula F is semantically equivalent to 0, i.e. $F \equiv 0$, then F is called unsatisfiable. If a formula F is semantically equivalent to 1, i.e. $F \equiv 1$, then F is called a tautology. ✍

Similarly, we introduce additional connectives as abbreviations:

$$A \oplus B :\equiv ((A \wedge (\neg B)) \vee ((\neg A) \wedge B))$$
$$A \leftrightarrow B :\equiv (A \wedge B) \vee ((\neg A) \wedge (\neg B))$$

Semantic equivalence is an equivalence relation: It partitions the set of all formulas into disjoint subsets, groups of semantically equivalent formulas, the equivalence classes. It structures the set of all syntactically correct formulas $\mathcal{E}_\mathcal{D}$ as visualized in the following diagram:



Number of Equivalence Classes The set of all syntactically correct formulas $\mathcal{E}_\mathcal{D}$ is infinite. The number of equivalence classes, however, is finite (if the set of atoms, $\mathcal{D}$, is). Let n be the number of atoms in $\mathcal{D}$, i.e. $|\mathcal{D}| = n$. Then there are 2^n different input configurations, i.e. the truth table has 2^n rows. As each row specifies an entry (0 or 1) of the truth vector, there are

$$\boxed{2^{2^n}}$$

different semantic behaviors, i.e. equivalence classes (Section 3.2.3).

We express the semantic equivalence of two formulas through a property of one single formula:

Theorem 2 (Relationship between semantic equivalence and tautology)**.** Two formulas F and G are semantically equivalent (Def. 6), i.e. $F \Leftrightarrow G$, if and only if the formula $F \leftrightarrow G$ is a tautology (Def. 7). ◁

Proof. The formula $F \leftrightarrow G \equiv (F \wedge G) \vee ((\neg F) \wedge (\neg G))$ is a tautology if and only if F and G have the same truth values for all assignments. Thus, it is a tautology if and only if F and G are semantically equivalent. □

Note that $\leftrightarrow$ connects F and G syntactically, whereas $\Leftrightarrow$ connects the two formulas semantically.

2.3 Logical Laws

So far, we have been sticking closely to the syntax permitted by Def 3. We introduce some simplifications of notation, motivated by equivalences that can be derived from the logical laws:

- We allow the connectives $(\oplus, \rightarrow, \leftrightarrow)$, as they are equivalent to formulas with basic connectives.
- If brackets do not change the truth behavior, they can be dropped. We write

$$\bigwedge_{i=1}^{n} A_i \equiv A_1 \wedge \ldots \wedge A_n, \quad \bigvee_{i=1}^{n} A_i \equiv A_1 \vee \ldots \vee A_n$$

- We introduce the priority rules (operator precedence):

$$\neg, (\wedge, \vee), (\oplus, \leftarrow, \rightarrow, \leftrightarrow)$$

  or if we want to be more extreme:

$$\neg, \wedge, \vee, \oplus, (\leftarrow, \rightarrow), \leftrightarrow$$

Note that the distributivity for AND and XOR does not hold if the two connectives are swapped (just as in arithmetic, there is a distributive law: $a(b + c) = ab + ac$, but not $a + bc = (a + b)(a + c)$ in general). In fact, XOR and AND can be sees as addition and multiplication of logic.

Equivalence	Name
$p \wedge \texttt{true} \equiv p$ $p \vee \texttt{false} \equiv p$	Identity laws
$p \vee \texttt{true} \equiv \texttt{true}$ $p \wedge \texttt{false} \equiv \texttt{false}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \wedge (q \oplus r) \equiv (p \wedge q) \oplus (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \texttt{true}$ $p \wedge \neg p \equiv \texttt{false}$	Negation laws
$p \to q \equiv \neg p \vee q$ $p \to q \equiv \neg q \to \neg p$ $p \vee q \equiv \neg p \to q$ $p \wedge q \equiv \neg(p \to \neg q)$ $\neg(p \to q) \equiv p \wedge \neg q$ $(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$ $(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$ $(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$ $(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$	Implication laws
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$ $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$ $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$ $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$	Biconditional laws

2.4 Normal forms

Definition 8 (Literal)**.** If $A \in \mathcal{D}$ is an atom, then A and $\neg A$ are called literals: A literal is an atom or a negated atom. ✍

Definition 9 (Disjunctive Normal Form)**.** A formula F is in disjunctive normal form (DNF) if there exist literals $L_{i,j}$ such that

$$F = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} L_{i,j} \right) = (L_{1,1} \wedge \ldots \wedge L_{1,m_1}) \vee \ldots \vee (L_{n,1} \wedge \ldots \wedge L_{n,m_n}) \quad ✍$$

Definition 10 (Conjunctive Normal Form)**.** A formula F is in conjunctive normal form (CNF) if there exist literals $L_{i,j}$ such that

$$F = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} L_{i,j} \right) = (L_{1,1} \vee \ldots \vee L_{1,m_1}) \wedge \ldots \wedge (L_{n,1} \vee \ldots \vee L_{n,m_n}) \quad ✍$$

Note that, in these definitions, we put equalities ($=$), not merely equivalences ($\equiv$): The normal forms are also syntactic notions.

Given a syntactically correct formula, one obtains its semantic behavior by writing down the truth table. It is also possible to construct a syntactically correct formula that reproduces a given truth table by considering either all the true rows (using Def 9) or all the false rows (using Def 10). The DNF and CNF obtained for a given formula F are syntactically different but, due to construction - we started out from the same truth vector - semantically equivalent.

2.5 Models and semantic conclusion

Definition 11 (Model)**.** Let F be a formula and $\mathcal{A}$ an assigment (Def 4) which renders F true. Then $\mathcal{A}$ is a model of F. We write $\mathcal{A} \models F$. ✍

Some of the properties of models are the following:

- Two formulas F and G are semantically equivalent, i.e. $F \equiv G$, if and only if they have the same models:
$$\mathcal{A} \models F \iff \mathcal{A} \models G$$
- A formula F is a tautology if and only if every assignment $\mathcal{A}$ is a model for F (F is true in any "thinkable" world).
- A formula F is unsatisfiable if and only if it has no model.

Based on the notions of models, we define the one-sided variant of semantic equivalence.

Definition 12 (Semantic Conclusion)**.** G is a semantic conclusion of $F_1, \ldots, F_n$ if every model $\mathcal{A}$ of all the F_i is also a model of G. We say that $F_1, \ldots, F_n$ semantically implies G. We write $\{F_1, \ldots, F_n\} \vDash G$ or $\{F_1, \ldots, F_n\} \Rightarrow G$. ✍

Semantic conclusion is the one-sided variant of semantic equivalence. They relate just like $\rightarrow$ and $\leftrightarrow$ on the syntactic level. In particular, two formulas F and G are semantically equivalent if and only if F is the semantic conclusion of G, and vice versa.

Note that the symbol $\models$ has two meanings: It indicates semantic conclusion as well as models.

Theorem 3 (Relationship between semantic implication and tautology)**.** A set of formulas $\{F_1, \ldots, F_n\}$ semantically implies a formula G, i.e. $\{F_1, \ldots, F_n\} \models G$, if and only if the formula $\bigwedge_{i=1}^{n} F_i \rightarrow G$ is a tautology. ◁

We carefully distinguish between the syntactical and the semantic levels: The expression $\{F_1, \ldots, F_n\} \vDash G$ relates the set of formulas $\{F_1, \ldots, F_n\}$ semantically with the formula G. The expression $\bigwedge_{i=1}^{n} F_i \rightarrow G$ connects the two syntactically and yields another syntactically correct formula. Only by demanding this formula to be a tautology does a semantic criterion emerge.

Proof. The implication $A \rightarrow B$ is equivalent to $\neg A \vee B$. Therefore,

$$\begin{aligned} \bigwedge_{i=1}^{n} F_i \rightarrow G &\equiv \neg \left(F_1 \wedge \ldots \wedge F_n\right) \vee G \\ &\equiv \neg F_1 \vee \ldots \vee \neg F_n \vee G \end{aligned}$$

This formula being a tautology means: If an assignment renders all F_i true - $\mathcal{A}(F_i) = 1 \; \forall i$ - then the assignment must also make G true. But that is exactly the definition of a semantic conclusion. □

Remark 1. A formula F is a tautology if and only if F is a conclusion of 1, i.e., $1 \vDash F$: F is a tautology if and only if the implication $1 \rightarrow F$ is a tautology.

A formula F is unsatisfiable if and only if 0 is a conclusion of F, i.e., $F \vDash 0$: F is unsatisfiable if the implication $F \rightarrow 0$ is a tautology. ◀

2.6 Proof theory of propositional logic

The goal of a proof theory is to decide semantic questions such as

- Is a formula F a tautology?
- Is a formula F unsatisfiable?
- Does $\{F_1, \ldots, F_n\} \models G$ hold?

In the end, this is always the same type of questions, that boils down to the question of whether a certain formula is a tautology or not.

Example 1 (Tautology Problem of CNF)**.** Given a formula F in CNF (Def 10), how can we decide wether it is a tautology? Since F is a conjunction of subformulas F_i, which are themselves disjunctions of literals:

$$F = \underbrace{(L_{1,1} \vee \ldots \vee L_{1,m_1})}_{=:F_1} \wedge \ldots \wedge \underbrace{(L_{n,1} \vee \ldots \vee L_{n,m_n})}_{=:F_n} = F_1 \wedge \ldots \wedge F_n$$

The formula F is a tautology if and only if all F_i are. The F_i are tautologies if at least one atom appears twice therein, once negated and once not negated. (Otherise, a violating assignment can always be found.) ◀

Example 2 (Tautology Problem of DNF)**.** Can we derive a similarly simple criterion for a formula F in DNF (Def 9)?

$$F = \underbrace{(L_{1,1} \wedge \ldots \wedge L_{1,m_1})}_{=:F_1} \vee \ldots \vee \underbrace{(L_{n,1} \wedge \ldots \wedge L_{n,m_n})}_{=:F_n} = F_1 \vee \ldots \vee F_n$$

Unfortunately, the problem does not "localize" (reduce to similar questions independently for the subformulas) in the same sense: The relation between subformulas is important here. Clearly, what can always be done is a truth table. However, the size of that grows exponentially with the number of different atoms in the formula. ◀

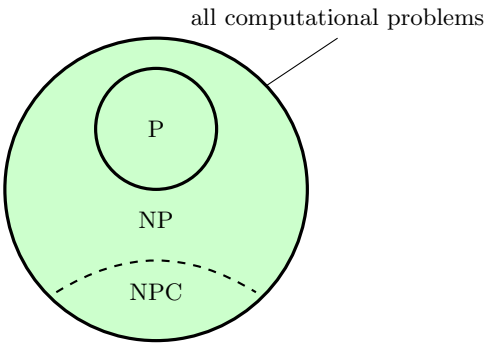
Relating DNF and CNF Is it easy to change from one normal form to the other, for a formula F? No, but what we can do is to obtain the negation of F in the other normal form in which F itself is given, using de Morgan's law:

$$\neg F = \neg \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} L_{i,j} \right) \equiv \bigwedge_{i=1}^{n} \left(\neg \bigwedge_{j=1}^{m_i} L_{i,j} \right) \equiv \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \neg L_{i,j} \right)$$

As F being a tautology is equivalent to $\neg F$ being unsatisfiable, the satisfiability problem for a DNF can be solved analogously to the tautology problem of a CNF: They are both simple. The other problems (satisfiability for CNF, tautology for DNF, change from CNF to DNF and vice versa for some F) seem hard. Let us look at this is some more detail.

Comparison of computational hardness Let us compare the hardness of the tautology problems for CNF and DNF, respectively: If F is given in CNF, one merely has to check the double occurrence of an atom (once positive, once negated) in each of the subformulas. This can be done in essentially linear time, i.e., the number of steps being upper-bounded by a linear function in the length of the formula, simply going through the formula. On the other hand, in the case of F being a DNF, one has to write down the entire truth table, containing 2^l rows. The number of computational steps is thus exponential in the length l, growing much faster. In a sense. the two problems are opposite extremal cases of hardness. Let us look at that in some more depth.

2.6.1 Short excursion into complexity theory



Complexity theory is about classifying computational problems by their computational difficulty. The set P contains all problems that can be solved in polynomial time, i.e., the number of computational steps being at most a polynomial function of the input size. P is a subset of NP, containing all problems for which one can verify a given solution in polynomial time, whereas the solution may not be found in polynomial time.

A subset of NP are the NP-complete problems (NPC): Problems in NPC are as hard as any other problem in NP. That means that any problem in NP can be reduced to any problem in NPC in polynomial time. Consequently, any problem in NPC can be reduced to another problem in NPC in polynomial time. This means that NPC problems can be used as indicators of simplicity for all NP problems: If you can solve one NPC problem in polynomial time, then you can solve all of them. (In fact, you have then shown P = NP = NPC, which would be an overly surprising result that makes you very famous; most people believe that the three classes P, NP, and NPC are all different from each other - no one has proven that so far, however.)

Example 3 (Elements of P)**.** P contains

- the tautology problem for CNF;
- the (un)satisfiability problem for DNF.

Even more, these decision problems can be solved not only in polynomial but even in linear time; they are among the simplest even among P problems. ◀

Example 4 (Elements in NPC)**.** NPC contains

- the tautology problem for DNF;
- the satisfiability problem for CNF
- the problem to find a semantically equivalent DNF for a given CNF, and vice versa.

The last follows from the first two: If we could find a G in CNF for any F in DNF with $F \equiv G$ in polynomial time, we could solve the tautology problem for a DNF in polynomial time by first turning it into a CNF and then applying the criteria above. ◀

Real-life problems? Are problems in real life hard? Yes, they are, as one can see in the following example.

Example 5 (Sudoku)**.** Sudoku is an NPC problem, as it boils down to the satisfiability of a CNF: As one has to find a solution satisfying all conditions on the rows, columns, and subboxes, it is a problem of the form

$$(\text{condition 1}) \wedge (\text{condition 2}) \wedge \dots \tag{1}$$

Each of these conditions can be satisfied in different ways:

$$\text{condition } i = \ \text{way } 1 \vee \ \text{way } 2 \vee \ \text{way } 3 \vee \dots$$

This is a CNF; to find a solution is the same as proving satisfiability. The satisfiability problem for a CNF is in NPC; so is Sudoku. ◀

Actually, this is a common structure of real-life problems. Usually there is a number of necessary conditions yielding a formula of the form (1). Each condition can be met in various ways, so the subformulas are disjunctions. Thus, one is usually looking for an assignment satisfying a CNF.

2.7 The resolution calculus

Although there is no hope to solve satisfiability (SAT) for CNF formulas efficiently always, we discuss here a calculus that can do it often.

Definition 13 (Resolution Step)**.** Let C_1, C_2, and R be clauses. R is the resolvent of C_1 and C_2 if there exists a literal L such that L is in C_1 and $\neg L$ is in C_2 and R contains all literals in C_1 and C_2 except of L and $\neg L$. In set notation:

$$R = C_1 \backslash \{L\} \cup C_2 \backslash \{\neg L\}.$$

The rationale is that C_1 and C_2 semantically imply R

$$\{C_1, C_2\} \vDash R.$$ ✍

Resolution does not merely show unsatisfiability but also yields an assignment if the formula is satisfiable.

That is, if a formula is not unsatisfiable, then the resolution calculus leads us to a satisfying assignment. A functional programming language directly based on resolution calculus is PROLOG.

3 Set Theory

Let us turn to the second pillar of mathematics, namely, set theory. As mentioned in the Introduction, all objects in mathematics are sets. For instance, the natural numbers can be defined inductively, starting from the empty set, through never-ending formation of new sets:

$$\begin{aligned} 0 &:= \emptyset \\ 1 &:= \{0\} = \{\emptyset\} \\ 2 &:= \{0,1\} = \{\emptyset, \{\emptyset\}\} \\ &\vdots \\ n &:= \{0,1,\ldots,n-1\} \end{aligned}$$

3.1 Basic notions

A *set* is a collection of objects - where all these objects are sets themselves (remember: all objects are). Thus, set theory is about a relation, called the element relation, among sets.

Definition 14 (Element Relation)**.** The relation "is an element of" relates an object (set) x[1] with a set A. One says "x is an element of A" or "x is in A" or "A contains x" and writes

$$x \in A.$$

If x is not in A, one writes

$$x \notin A \quad \text{or} \quad \neg(x \in A). \qquad \text{✍}$$

3.1.1 Cantor's Paradise

Georg Cantor is the founder of set theory. His definition of what a set is, and what objects it can contain, was very liberal in the sense that a set was a collection of any kind of objects with the only condition that these objects be distinguishable from each other.

Definition 15 (Cantor's "naïve" approach)**.** Any collection of well-distinguished objects is a set. ✍

In this liberal definition, it is, in particular, not excluded that a set contains itself as a member. Unfortunately, this possibility leads to a serious problem: Consider the set

$$M := \{B \mid B \notin B\}\,.$$

Does M contain itself? If it does, it does not, according to the definition—a classical antinomy: Naïve set theory crashes. This antinomy, which is due to the mathematician and philosopher Bertrand Russell, was, by Russell himself, put as follows: A barber is a man who shaves every man who does not shave himself. Does the barber shave himself? Again, he does exactly when he does not.

That was the end of Cantor's Paradise. The way out is a set of stricter rules about what can be a set. It will still be the case that sets contain sets (there is nothing else, after all), but with the new rules, there is no more such a thing as "the set of all sets". And in particular, it never occurs that a set contains itself. The most famous set of such rules goes back to Ernst Zermelo and Abraham Fränkel (ZF), extended by the somewhat mysterious "axiom of choice" (ZFC).

[1] It is common to label sets with capital letters and their elements with small letters - although, again, also the lowercase-letter objects are sets.

3.1.2 Zermelo/Fraenkel/Choice (ZFC) set theory

is a set of axioms describing how sets can be formed. First, we introduce symbols from predicate logic which we use as linguistic abbreviations.

Definition 16 (Quantifiers)**.** The universal quantifier $\forall$ is used to express that a statement holds for all objects of a certain kind. The existential quantifier $\exists$ expresses that a statement holds for at least one object. ✍

Axiom 17 (Extensionality)**.** Two sets are equal if they have the same elements:

$$\forall A \, \forall B \, (\forall x \, (x \in A \Leftrightarrow x \in B) \Rightarrow A = B).$$ ✍

While Axiom 17 states a sufficient condition, the necessity is logical: The logic of equality states that if two objects are equal, they have the same properties.

Definition 18 (Predicate)**.** A predicate on a set A is a function $P\colon A \to \{\texttt{false}, \texttt{true}\}$. P can also be seen as a property that all $x \in A$ have for which $P(x) = \texttt{true}$. ✍

Axiom 19 (Subsets from Predicates)**.** Given a set A and a predicate $P\colon A \to \{\texttt{false}, \texttt{true}\}$, the collection

$$\{x \in A \mid P(x)\} := \{x \in A \mid P(x) = \texttt{true}\}$$

is another set. ✍

Definition 20 (Subset)**.** $A \subseteq B :\Longleftrightarrow \forall x \, (x \in A \Rightarrow x \in B)$ ✍

Theorem 4. If $A \subseteq B$ and $B \subseteq A$, then $A = B$. ◁

Proof. Consequence of Definition 20 and Axiom 17. □

The existence of an empty set is usually postulated. One may also obtain it from any non-empty set A using Axiom 19 with predicate $x \neq x$:

$$\emptyset := \{x \in A \mid x \neq x\}$$

Out of this semen alone, the whole rich zoo of mathematical objects blossoms.

Definition 21 (Intersection)**.** $x \in A \cap B :\Longleftrightarrow x \in A \wedge x \in B$ ✍

Axiom 22 (Union)**.** $x \in A \cup B :\Longleftrightarrow x \in A \vee x \in B$ ✍

Definition 23 (Difference)**.** $x \in A \setminus B :\Longleftrightarrow x \in A \wedge x \notin B.$ ✍

Definition 24 (Symmetric difference)**.** $x \in A \triangle B :\Longleftrightarrow (x \in A) \oplus (x \in B).$ ✍

$A \triangle B$ can also be written as $(A \setminus B) \cup (B \setminus A)$ or $(A \cup B) \setminus (A \cap B)$.

The definitions above show the close relation between set theory and logic: Using the logical connectives, we can define corresponding set operations.

The construction of the power set goes beyond that, and it is the most "powerful" of the ZFC set-forming axioms, allowing for constructing in particular very large sets from smaller ones.

Axiom 25 (Power Set)**.** For every set A the power set

$$\mathbb{P}(A) := \{X \mid X \subseteq A\}$$

is a set. In particular, it contains the empty set $\emptyset$ and A itself. ✍

The power set is also denoted 2^A because the cardinality (i.e. the number of elements) of the power set is

$$|\mathbb{P}(A)| = 2^{|A|}$$

if A is finite. This is since for each of the $|A|$ elements of A, we can decide whether we choose it or not for the subset: We have $|A|$ binary choices, multiplying up to $2^{|A|}$ total choices, i.e., different subsets.

Definition 26 (Complement)**.** Given a universe $\mathcal{U}$, the complement of $A \subseteq \mathcal{U}$ is

$$\overline{A} := \mathcal{U} \setminus A$$ ✍

For a family $\{A_i\}_{i \in I}$ we define

$$x \in \bigcup_{i \in I} A_i :\Longleftrightarrow \exists i \in I\,(x \in A_i), \qquad x \in \bigcap_{i \in I} A_i :\Longleftrightarrow \forall i \in I\,(x \in A_i)$$

Let us finally look at this mysterious "axiom of choice". Intuitively, it asks for a quite unsurprising fact: If you have a family of all non-empty sets, then it is possible to choose exactly one element out of each of the (non-empty) sets in the family. (Another way of putting it is: The Cartesian product of a family of non-empty sets is non-empty.) What is so mysterious about that? Although it is intuitive - why would the claimed not be possible? - it has counterintuitive consequences, such as the Banach/Tarski paradox: A unit ball can be cut into five peaces (subsets) that can be rearranged using only rotations and translations, for obtaining two unit balls of the same size. This suggests that our intuition is somehow inconsistent. In this sense, Banach/Tarski is only a paradox, and not an antinomy: It appears weird to us, but it is not an intrinsic logical problem of the theory. Which does not mean that it does not have consequences: For instance, it is not possible to define a universal volume function. The reason why the axiom of choice appears so innocent to us is that we apply it to finite families, whereas the strange consequences come when you apply it beyond this, to infinite families. Major parts of mathematics depend on that axiom, such as most of functional analysis, which is the basis for quantum mechanics.

Axiom 27 (Choice)**.** Given a family of sets, each containing at least one element, it is possible to make a selection of exactly one object from each set. ✍

3.1.3 Laws derived from logic

We can derive many laws from their logical counterparts by replacing the connectives by their set-theoretic counterparts in a logical law from Section 2.3: $\wedge \mapsto \cap$, $\vee \mapsto \cup$, $\neg \mapsto \bar{\cdot}$, $\texttt{true} \mapsto \mathcal{U}$, $\texttt{false} \mapsto \emptyset$, $\oplus \mapsto \triangle$, $\equiv \mapsto =$

3.1.4 The Cartesian product

In the 17$^{\text{th}}$ century, the early modern philosopher René Descartes introduced the Cartesian product to describe the location of points by their coordinates. Only later Cartesian products were formalized in set theory employing ordered pairs.

Definition 28 (Ordered pair)**.** $(x, y) := \big\{\{x\}, \{x, y\}\big\}$ ✍

Definition 29 (Cartesian Product)**.** $A \times B := \{\,(a, b) \mid a \in A \wedge b \in B\,\}$ ✍

The definitions of ordered pairs extends naturally to ordered lists of more than two numbers, so called tuples. Given a finite index set, $I = \{1, \ldots, k\}$ we define the Cartesian product of k sets as

$$\bigtimes_{i \in I} A_i := \big\{\,(a_1, \ldots, a_k) \mid \forall i \in I,\ a_i \in A_i\,\big\}$$

3.2 Relations

Definition 30 (Binary relation)**.** A (binary) relation R from A to B is a subset of the cartesian product of A and B, i.e. $R \subseteq A \times B$. We write $a\ R\ b$ for $(a, b) \in R$. ✍

Since they are sets, we can apply set calculus on relations.

Example 6. For the relations $\leq$, $\geq$, $<$, $>$, $=$, $\neq$ we have

$$\leq \cap \geq\ =\ =, \quad \leq \triangle \geq\ =\ \neq, \quad < \subseteq \leq, \quad > \subseteq \geq, \quad \overline{\leq}\ =\ >, \quad \overline{\geq}\ =\ <$$ ◀

Example 7 (Divisibility, Modulo)**.** An integer b is said to be divisible by another integer a if there exists an integer $c \in \mathbb{Z}$ such that $a \cdot c = b$, i.e.

$$a \mid b \quad :\Longleftrightarrow \quad \exists c \in \mathbb{Z} : a \cdot c = b$$

It is a binary relation on $\mathbb{Z}$, i.e. $\mid \subseteq \mathbb{Z} \times \mathbb{Z}$. A binary relation on $\mathbb{R}$ can be reduced to a relation on $\mathbb{Z}$ by interesection:

$$R_{\mathbb{Z}} = R_{\mathbb{R}} \cap \mathbb{Z}^2$$

Congruence modulo m is defined as

$$a \equiv b \pmod{m} \quad :\Longleftrightarrow \quad m \mid (a - b)$$

This is the same as saying that the integer division of each a and b by m yields the same remainder. Each natural number m defines a congruence relation $\equiv_m$ with the modulo m. The intersection of two such congruence relations yields another one with the least common multiple[2] being the modulus:

$$\equiv_{m_1} \cap \equiv_{m_2} = \equiv_{\mathrm{lcm}(m_1, m_2)}$$ ◀

3.2.1 Representation of relations

Relations on finite sets A and B can be represented by either binary matrices or bipartite graphs. In matrix representation, each row corresponds to an element in A and each column to an element in B. The entry is then 1 if and only if the corresponding pair (a, b) is in R. In the bipartite graph, the nodes on the left correspond to elements in A, the ones on the right to elements in B. The nodes are linked if and only if $(a, b) \in R$.

In the special case of relations on a set A, i.e., where $A = B$ holds, representations can be made so that every element a of A is drawn only once, and an arrow connects a_1 and a_2 if and only if the pair (a_1, a_2) is in the relation. What results is a graph. It is, however, not a simple graph since it can have loops, i.e., a node connected by itself via an arrow. In some way, a graph is in fact nothing but (the representation of) a relation on its vertex set.

3.2.2 Properties of relations

For a relation $R \subseteq A \times A$:

- Reflexive: $\forall a : (a, a) \in R$. In matrix representation, the corresponding matrix has all 1 on the diagonal. The representing graph has a loop for each of its nodes.
- Anti-reflexive: $\forall a : (a, a) \notin R$. The diagonal of the corresponding matrix is 0, and there are no loops in the associated graph.
- Symmetric: $\forall a, b : (a, b) \in R \Rightarrow (b, a) \in R$. For finite sets the corresponding binary matrix is symmetric. In the graph representation, it means that whenever there is an arrow in one direction, there is also one in the inverse direction, i.e. the graph is undirected.
- Anti-symmetric: $\forall a, b : (a, b) \in R \wedge (b, a) \in R \Rightarrow a = b$. In the matrix representation, whenever a non-diagonal position (i, j) holds a 1 , then the mirrored position (j, i) must hold a 0. A 0 in both positions is possible.
- Transitive: $\forall a, b, c : (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$. This is the central property shared by equivalence and order relations.

[2] The least common multiple of two integers a and b is the smallest number $c \in \mathbb{Z}$ such that there exist integers $m, n \in \mathbb{Z}$ with $a \cdot m = c = b \cdot n$. For relatively prime numbers a and b, the least common multiple is simply their product, $c = a \cdot b$.

3.2.3 Equivalence relations

Definition 31 (Equivalence relation)**.** A relation $\sim$ on a set A is called an equivalence relation if it is reflexive, symmetric, transitive. See 3.2.2. ✍

Definition 32 (Partition)**.** A partition of a set A is a family of sets $(A_i)_{i\in I}$ such that

$$\bigcup_{i\in I} A_i = A \qquad \text{and} \qquad A_i \cap A_j = \emptyset \quad \forall i, j \in I,\ i \neq j$$ ✍

Theorem 5. Any equivalence relation yields a partition, and vice versa. More precisely, if $\sim$ is an equivalence relation on A, then the equivalence classes

$$[a] := \{x \in A \mid x \sim a\} \subseteq A$$

are a partition of A. Conversely, if $(A_i)_{i\in I}$ is a partition of A, then the relation

$$x \sim y \quad :\Leftrightarrow \quad \exists i \in I : x \in A_i \wedge y \in A_i$$

is an equivalence relation. ◁

Example 8 (Congruence modulo m)**.** The relation $\equiv_m$ from Example 7 is an equivalence relation on $\mathbb{Z}$ for any $m \in \mathbb{Z}$ with m equivalence classes. The set of equivalence classes is denoted by $\mathbb{Z}_m := \mathbb{Z}/\equiv_m = \{[0], \ldots, [m-1]\}$. On $\mathbb{Z}_m$, we can define

$$[a] + [b] := [a + b], \qquad [a] \cdot [b] := [a \cdot b],$$

yielding $(\mathbb{Z}_m, +, \cdot)$, an algebraic structure called a ring. ◀

3.2.4 Order relations

Definition 33 (Partial Order)**.** A relation $\preccurlyeq$ on a set A is called a partial order if it is reflexive, anti-symmetric, transitive. See 3.2.2. ✍

Example 9. Examples of partial orders include:

- On $\mathbb{N}, \mathbb{Z}, \mathbb{R}$, $\leq$ and $\geq$ are partial orders, but not $<$ or $>$, because they are not reflexive.
- Divisibility $\mid$ on $\mathbb{N}$, but not on $\mathbb{Z}$, because it is not reflexive there.
- The subset relation $\subseteq$ on a power set $\mathbb{P}(B)$ is a partial order. ◀

In a poset $(A, \preccurlyeq)$ an element x is

- a maximal element if $\nexists y \in A\ (y \neq x \wedge x \preccurlyeq y)$
- the greatest element if $\forall y \in A,\ y \preccurlyeq x$

Note that for a partial order, we do not require that every pair of elements is comparable. This condition

$$\forall x, y \in A : x \preccurlyeq y \vee y \preccurlyeq x$$

is only required for a total order, also called linear order. For a total order, in the directed graph representation, all nodes are connected.

3.3 Functions

Definition 34 (Function)**.** A relation $f \subseteq A \times B$ is a function(al relation) from A to B, written $f\colon A \to B$, if it satisfies the properties

(i) $\forall a \in A\ \exists b \in B :\ (a,b) \in f$ (existence of functional value)

(ii) $(a,b) \in f \wedge (a,b') \in f \Rightarrow b = b'$ (uniqueness)

(i) and (ii) are sometimes expressed together as $\forall a \in A\ \exists! b \in B :\ (a,b) \in f$. ✍

The definition identifies a function with the corresponding set of pairs (a,b), i.e., the graph of the function. The following notations are commonly used for $(a,b) \in f$:

$$f(a) = b, \quad f : a \mapsto b$$

Definition 35 (injective, surjective, bijective)**.** Let $f\colon A \to B$.

- f is injective if $f(a) = f(a') \Rightarrow a = a'$
- f is surjective if $\forall b \in B\ \exists a \in A,\ f(a) = b$
- f is bijective if it is both injective and surjective ✍

The properties from Definition 35 can be used to introduce a relation on sets and compare them with respect to their size (cardinality).

Definition 36 (Cardinality)**.** For sets A, B:

$$A \preccurlyeq B :\Longleftrightarrow \exists \text{ injective } f\colon A \to B, \qquad A \approx B :\Longleftrightarrow \exists \text{ bijective } f\colon A \to B$$ ✍

The relation $\preccurlyeq$ is a total order. The following theorem shows that $\preccurlyeq$ has a property resembling anti-symmetry (see 3.2.2).

Theorem 6 (Cantor/Schröder/Bernstein)**.** If $A \preccurlyeq B$ and $B \preccurlyeq A$, then $A \approx B$. ◁

Proof. Imagine a park ($= A$) with a house ($= B$). Inside the house, there is a map of the park. If we look closely, we see the house on the map again. This house on the map in turn contains a map fo the park containing a house containing a map of the park, and so on, ad infinitum. As the house is inside the park, i.e. $B \subseteq A$, there exists an injective function $g : B \to A$. On the other hand, the map of the park is inside the house, i.e. $A \subseteq B$, and there exists an injective function $f : A \to B$. We define a bijection $h\colon A \to B$ as follows: Points in the set "park without house" ($= A \setminus B$) are mapped to their correspondend an iteration level below. Points in the set "house without map" ($= B \setminus A$) are mapped to themselves. □

Theorem 7 (Cantor)**.** For any set A we have $A \prec \mathbb{P}(A)$; i.e. the power set of A is strictly larger than A itself. ◁

Proof. Assume $f\colon A \to \mathbb{P}(A)$ were surjective and define set $B := \{a \in A \mid a \notin f(a)\} \subseteq A$. There is no $b \in A$ with $f(b) = B$: if $b \in B$, then $b \notin f(b) = B$; if $b \notin B$, then $b \in f(b) = B$. Hence f is not surjective. □

The results above open the door to ever larger infinities: $\mathbb{R}$ has the same size as $\mathbb{P}(\mathbb{N})$, which in turn is dwarfed by $\mathbb{P}(\mathbb{R})$, and so on ad infinitum.

4 Combinatorics

is a collection of methods, principles, tools, techniques, and facts to count the size of finite sets with some structure.

4.1 Binomial Coefficient and Pascal's Triangle

Definition 37 (Binomial Coefficient)**.**

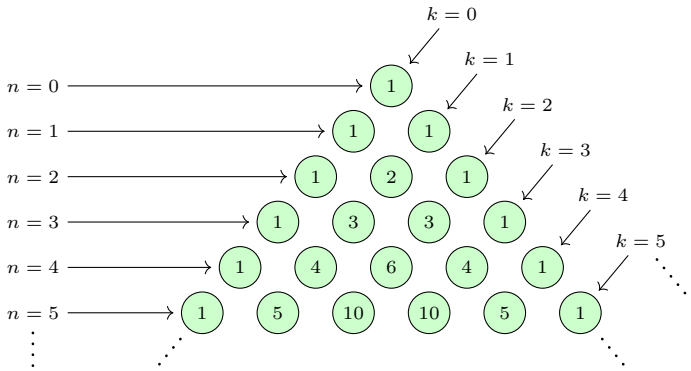
$$\binom{n}{k} := \frac{n!}{k!(n-k)!} = \binom{n-1}{k-1} + \binom{n-1}{k} \tag{2}$$

with the base cases

$$\binom{n}{0} := \binom{n}{n} := 1 \tag{3}$$

We say "n choose k" referring to $\binom{n}{k}$, because it is the number of ways in which we can choose k out of n elements (Unordered selection without repetition). In terms of sets, it is the number of subsets of size k of a set of size n. Let S be an n-element set and fix $x \in S$. A k-subset of S either excludes x (yielding $\binom{n-1}{k}$ choices) or includes x (yielding $\binom{n-1}{k-1}$ choices, obtained by selecting the remaining $k-1$ elements from $S \setminus \{x\}$). Hence the number of k-subsets of S is $\binom{n-1}{k} + \binom{n-1}{k-1}$, yielding the recursion relation on the right side in (2).

The value of $\binom{n}{k}$ is completely determined by the right side in (2), as it can be computed for any n and k by building up the following triangle employing the recursion relation in (2), starting with the base cases in (3).



Algorithmically speaking, simply applying the recursion in (2) leads to very inefficient computation. This is because the smaller subproblems to which the original problem is reduced are solved repeatedly, even if they have already been computed. Computing the coefficient would amount to adding up 1's. The remedy when naive recursion fails in this way is dynamic programming: filling a table with intermediate results (in our case, the triangle) to avoid redundant computations.

4.2 Urn Model

4.2.1 Permutations (arrange)

Arrangement without repetition. The number of possibilities to arrange n distinct elements is

$$\boxed{P_n = n!} \tag{4}$$

Arrangement with repetition. The number of ways to arrange n repeated elements that are of l distinct groups with multiplicities $m_1, \ldots, m_l$ is

$$\boxed{P_{m_1,\ldots,m_l} = \frac{\left(\sum_{i=1}^{l} m_i\right)!}{\prod_{i=1}^{l} (m_i!)} = \frac{n!}{m_1! \cdots m_l!}} \tag{5}$$

4.2.2 Variations (choose and arrange)

Ordered selection with repetition. After each draw, the element is put back. For each of the k draws there are n choices, so the number of k-tuples is

$$\boxed{\overline{V}_k^n = n^k} \tag{6}$$

Ordered selection without repetition. Elements once drawn are not put back. The number of k-tuples is

$$\boxed{V_k^n = \frac{n!}{(n-k)!} =: n^{\underline{k}}} \tag{7}$$

4.2.3 Combinations (choose)

Unordered selection without repetition. Here we simply choose k elements out of n without caring for order, hence the number of combinations is

$$\boxed{C_k^n = \frac{n!}{k! \cdot (n-k)!} \overset{(2)}{=:} \binom{n}{k}} \tag{8}$$

Unordered selection with repetition. Here we choose k elements from n elements, but we are allowed to choose the same element multiple times. This is equivalent to putting k indistinguishable balls into n distinguishable boxes, where each box can hold any number of balls. We can think of the k balls as k stars, and the $n-1$ dividers between the boxes as $n-1$ bars:

$$\star \ \star \ \cdots \ \star \ \big| \ \star \ \cdots \ \star \ \big| \ \cdots \ \big| \ \star \ \cdots \ \star$$

The combinations are characterized merely by the order of the stars and bars, so the number is given by the number of ways to arrange k indistinguishable stars and $n-1$ indistinguishable bars (Arrangement with repetition).

$$\boxed{\overline{C}_k^n = \underbrace{\frac{(n+k-1)!}{k! \cdot (n-1)!}}_{\hat{=} P_{k,n-1}} = \underbrace{\binom{n+k-1}{k}}_{\hat{=} C_k^{n+k-1}}} \tag{9}$$

It is also equivalent to distributing k indistinguishable balls into $k+n-1$ distinguishable spots, where each spot can hold at most one ball. This is the same as choosing k spots from $n+k-1$ available spots (Unordered selection without repetition).

4.3 Rules and strategies

When faced with a counting problem, it is often possible to count parts of the set using the standard urn models (4.2). What remains is then the problem of composing the partial solutions. We recall some basic principles.

Sum Rule. For a family $(A_i)_{i=1,\dots,n}$ of mutually disjoint sets,

$$\left| \bigcup_i A_i \right| = \sum_i |A_i|$$

Product Rule. For a family $(A_i)_{i=1,\dots,n}$ of sets (may not be disjoint),

$$\left| \bigtimes_i A_i \right| = \prod_i |A_i|$$

Equality Rule. Finite sets in bijection have the same number of elements.

4.3.1 Inclusion-Exclusion Principle.

We generalize the Sum Rule to sets that are not mutually disjoint.

Theorem 8 (Inclusion/Exclusion)**.** For a family $(A_i)_{i=1,\dots,n}$ of sets (may not be disjoint),

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{r=1}^{n}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n} \left|\bigcap_{k=1}^{r} A_{i_k}\right| \tag{10}$$

The outer sum runs over all possible sizes r of intersections, from 1, i.e., the single sets, to n, i.e., the intersection of all sets. Therefore, for $r = 1$, the inner sum contains n terms of the form $|A_i|$, and for $r = n$, it contains only one term, $|A_1 \cap \ldots \cap A_n|$. In general, the inner sum contains $\binom{n}{r}$ terms, as it runs over all possible r-fold intersections of n sets, i.e. all possible ways of choosing r sets out of n sets (Unordered selection without repetition). ◁

Proof. Induction over n.

(i) **Base case.** The union of two sets ($n = 2$), disjoint or not, is of size

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

The rationale is: If we simply add the two sizes, then the "overlap", i.e., the elements that are in both sets, are counted twice; hence, we have to "remove them once". ✔

(ii) **Induction hypothesis.** Assume (10) holds for the union of n sets, for some $n \in \mathbb{N}$.

(iii) **Induction step.** Consider a union of a family of $(n+1)$ sets - of which we single out the $(n+1)^{\text{th}}$ set.

$$\left|\bigcup_{i=1}^{n+1} A_i\right| = \left|\left(\bigcup_{i=1}^{n} A_i\right) \cup A_{n+1}\right| \overset{\text{(i)}}{=} \left|\bigcup_{i=1}^{n} A_i\right| + |A_{n+1}| - \overbrace{\left|\left(\bigcup_{i=1}^{n} A_i\right) \cap A_{n+1}\right|}^{\overset{3.1.3}{=}\left|\bigcup_{i=1}^{n}(A_i\cap A_{n+1})\right|} \tag{11}$$

$$\overset{\text{(ii)}}{=} \sum_{r=1}^{n}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n} \left|\bigcap_{k=1}^{r} A_{i_k}\right| + |A_{n+1}| - \sum_{r=1}^{n}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n} \left|\bigcap_{k=1}^{r} (A_{i_k} \cap A_{n+1})\right| \tag{12}$$

$$= \sum_{k=1}^{n} |A_k| + \sum_{r=2}^{n}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n} \left|\bigcap_{k=1}^{r} A_{i_k}\right| + |A_{n+1}| - \sum_{r=1}^{n}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n} \left|\left(\bigcap_{k=1}^{r} A_{i_k}\right) \cap A_{n+1}\right| \tag{13}$$

$$= \sum_{k=1}^{n+1} |A_k| + \sum_{r=2}^{n+1}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n} \left|\bigcap_{k=1}^{r} A_{i_k}\right| + \sum_{r=2}^{n+1}(-1)^{r-1} \underbrace{\sum_{\substack{1\le i_1<\dots<i_{r-1}\le n \\ i_r=n+1}} \left|\left(\bigcap_{k=1}^{r-1} A_{i_k}\right) \cap A_{n+1}\right|}_{=\sum_{1\le i_1<\dots<i_{r-1}\le n}\left|\bigcap_{k=1}^{r} A_{i_k}\right|} \tag{14}$$

$$= \sum_{r=1}^{n+1}(-1)^{r-1} \sum_{1\le i_1<\dots<i_r\le n+1} \left|\bigcap_{k=1}^{r} A_{i_k}\right| \tag{15}$$

Note that we have used (i) in (11). Going from (11) to (12), we have applied (ii) twice; to $\left|\bigcup_{i=1}^{n} A_i\right|$ as well as $\left|\bigcup_{i=1}^{n} (A_i \cap A_{n+1})\right|$. In (12), we split off the case $r = 1$ from the first sum, yielding (13). In order to obtain (14) from (13), the index r in the last term was shifted and the emerging $1/(-1)$ was taken in front of the sum. Finally, to obtain (15) from (14), the terms containing $i_r = n + 1$ were combined with the terms not containing $i_r = n + 1$ and $\sum_{k=1}^{n+1} |A_k|$ was absorbed into the resulting sum as $r = 1$. ✔

This concludes the proof of the Inclusion/Exclusion Theorem. □

Example 10 (Opera)**.** During an opera with n guests, all the coats in the cloakroom get disordered, and every guest gets back a random coat afterwards. We want to find the probability that at least one guest gets back their own coat. If no guest receives their own coat, the corresponding permutation is fixed-point free.

We denote the set containing all permutations with at least one fix-point as A. We can express this set as a union of sets A_i, where A_i is the set of permutations for which i is a fixed point (there can be other fixed points). Then,

$$A = \bigcup_{i=1}^{n} A_i$$

and using the Inclusion/Exclusion principle, we can compute the size of A. At first sight it might seem unwise to reduce counting a single set to counting an exponential number of sets. Luckily, many of these sets have the same size, which is easy to compute and the number of sets in such groups is easy to determine as well.

The size of A_i is given by the number of permutations of the remaining $n-1$ elements, which is $(n-1)!$. Similarly, the size of r-fold intersections of the sets A_i is given by the number of permutations of the remaining $n-r$ elements, which is $(n-r)!$.

As mentioned in Theorem 8, the number of distinct r-fold intersections is given by $\binom{n}{r}$. So,

$$\begin{aligned} |A| = \left|\bigcup_{i=1}^{n} A_i\right| &= \sum_{r=1}^{n} (-1)^{r-1} \sum_{1 \le i_1 < \cdots < i_r \le n} \left|\bigcap_{k=1}^{r} A_{i_k}\right| \\ &= \sum_{r=1}^{n} (-1)^{r-1} \binom{n}{r} (n-r)! \\ &= n! \sum_{r=1}^{n} \frac{(-1)^{r-1}}{r!} \end{aligned}$$

The number of fixed-point free permutations is then

$$\#\,\mathrm{FPFP}(n) = n! - |A| = n! \sum_{r=0}^{n} \frac{(-1)^r}{r!}$$

where $\#\,\mathrm{FPFP}(n)$ denotes the number of fix-point free permutations of size n. For large n, we have $\#\,\mathrm{FPFP}(n) \approx \frac{n!}{e}$. The probability that at least one guest receives their own coat is thus given by

$$\begin{aligned} &\mathrm{P}(\text{at least one guest receives their own coat}) \\ &= 1 - \mathrm{P}(\text{no guest receives their own coat}) \\ &= 1 - \frac{\#\,\mathrm{FPFP}(n)}{n!} = 1 - \sum_{r=0}^{n} \frac{(-1)^r}{r!} \xrightarrow{n\to\infty} 1 - \frac{1}{e} \end{aligned}$$

What is remarkable is that the probability quickly converges to a constant value that is independent of the number of guests. ◀

4.3.2 The Pigeonhole Principle

"When n objects are distributed among k boxes with $k < n$, at least one box contains at least two objects."

Theorem 9 (Pigeonhole Principle)**.** If a set of n objects is partitioned into $k < n$ sets, then at least one of these sets contains at least

$$\left\lceil \frac{n}{k} \right\rceil$$

objects. ◁

Proof. Contradiction. Suppose that all sets in the partition have at most $\left\lceil \frac{n}{k} \right\rceil - 1$ objects. Then the total number of objects is at most $k\left(\left\lceil \frac{n}{k} \right\rceil - 1\right)$, which is smaller than n because

$$k\left(\left\lceil \frac{n}{k} \right\rceil - 1\right) < k\left(\left(\frac{n}{k} + 1\right) - 1\right) = k\left(\frac{n}{k}\right) = n$$

□

Proposition 10 (Erdős-Szekeres)**.** A sequence of $m^2 + 1$ distinct numbers contains a monotonic subsequence of length $m + 1$, and this bound is thight. ◁

Proof. Contradiction. Consider a sequence of $m^2 + 1$ elements

$$(a_1, \ldots, a_{m^2+1})$$

and assume that the longest subsequence has length $l \leq m$. To each element a_i, assign a pair $(c_i, d_i) \in \mathbb{N}^2$ representing the length of the longest monotonically increasing/decreasing subsequence starting with (and including) a_i. By the assumption, we have $c_i, d_i \leq m \ \forall i \in \{1, \ldots, m^2 + 1\}$. Thus, there are at most m^2 (Ordered selection with repetition) possible different ordered pairs (c_i, d_i). By the Pigeonhole Principle[3], at least two elements a_i and a_j must have the same pair (c, d). We can distinguish two cases:

(i) $a_i \leq a_j \Rightarrow c_i \overset{\lightning}{>} c_j$

(ii) $a_i \geq a_j \Rightarrow d_i \overset{\lightning}{>} d_j$

In both cases, (i) and (ii), the assumption leas to a contradiction. Hence, the longest monotonic subsequence must have length $l \geq m + 1$. The bound is also tight, as it is easy to construct a sequence of length m^2 of which the longest monotonic subsequence has length only m. □

4.3.3 Double counting

A relation $S \subseteq A \times B$ can be counted "row-wise" or "column-wise":

$$|S| = \sum_{a \in A} |\{b \mid (a, b) \in S\}| = \sum_{b \in B} |\{a \mid (a, b) \in S\}|$$

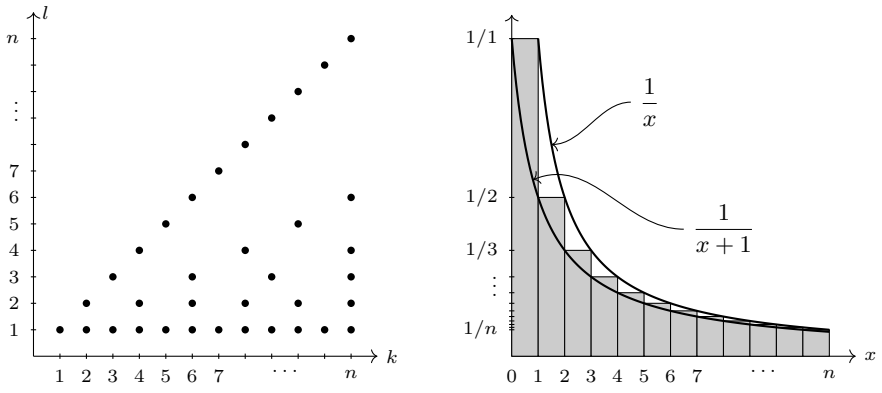
Example 11 (Average Number of Divisors)**.** We want to find the average number of divisors of numbers up to n. We introduce function counting the number of divisors of k:

$$\nu(k) := |\{l > 0 \mid l \text{ divides } k\}|$$

i.e. $\nu(1) = 1$, $\nu(2) = 2$, $\nu(3) = 2$, $\nu(4) = 3$, $\nu(5) = 2$, $\nu(6) = 4$, $\nu(7) = 2$, etc. Furthermore, for every prime number p, we obtain $\nu(p) = 2$. For a given number n, we are interested in calculating

$$\frac{1}{n} \sum_{k=1}^{n} \nu(k)$$

Summing up the values of $\nu(k)$ is equivalent to counting the dots in the left image column by column.



Whereas the columns have a irregular pattern, the rows are very regular: Every second number is even, every third is divisible by three, etc. In particular, the number of points

[3] The m^2 possible pairs correspond to the holes and the $m^2 + 1$ elements of the sequence to the pigeons.

within each row is given by $\lfloor \frac{n}{k} \rfloor$: The fraction $\frac{n}{k}$ is rounded to the next smaller integer value. Replacing the sum over columns by a sum over rows, we obtain

$$\frac{1}{n}\sum_{k=1}^{n}\nu(k) = \frac{1}{n}\sum_{l=1}^{n}\left\lfloor\frac{n}{k}\right\rfloor$$

Now, we want to estimate how this number grows with n. Note that $\frac{n}{l} - 1 \leq \lfloor \frac{n}{l} \rfloor \leq \frac{n}{l}$. Therefore, we can bound the average as

$$\left(\sum_{l=1}\frac{1}{l}\right) - 1 \leq \frac{1}{n}\sum_{l=1}^{n}\left\lfloor\frac{n}{l}\right\rfloor \leq \sum_{l=1}^{n}\frac{1}{l}$$

Thus, we require bounds on the sum $\sum_{l=1}^{n} 1/l$. As shown in the image on the right, we can approximate with functions and integrate to obtain the area below the graph.

The upper bound is then

$$\sum_{l=1}^{n}\frac{1}{l} \leq 1 + \int_{1}^{n}\frac{1}{x}\mathrm{d}x = 1 + \ln(n)$$

where $\ln(\cdot)$ is the natural logarithm. Similarly, we obtain the lower bound

$$\sum_{l=1}^{n}\frac{1}{l} \geq \int_{0}^{n}\frac{1}{x+1}\mathrm{d}x = \ln(n+1) \geq \ln(n)$$

Putting all this together, the average of the number of divisors for numbers between 1 and n can be estimated as

$$\ln(n) - 1 \leq \frac{1}{n}\sum_{l=1}^{n}\left\lfloor\frac{n}{l}\right\rfloor \leq \ln(n) + 1$$ ◀

4.4 Binomial Coefficients: Properties and Approximations

Recall the definition of the Binomial Coefficient (Definition 37).

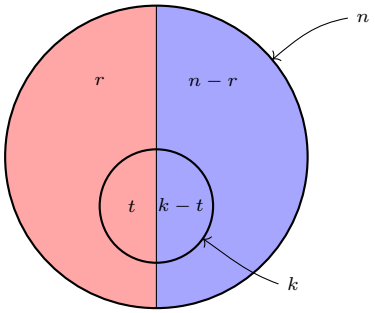
4.4.1 Symmetry

The binomial coefficient reflects the symmetry of the Pascal triangle as

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{(n-k)!k!} = \binom{n}{n-k} \tag{16}$$

4.4.2 Vandermonde Identity

Suppose an n-element set is partitioned into a red part of size r and a blue part of size $n - r$:



The number of possibilities to choose k elements such that t are red and $k - t$ is the product of the number of ways to choose t red elements from the red part and $k-t$ blue elements from the blue part:

$$\binom{r}{t} \cdot \binom{n-r}{k-t}$$

The total number of possibilities to choose k elements from the n-set is then the sum over all possible values of t:

$$\binom{n}{k} = \sum_{t=0}^{k} \binom{r}{t} \cdot \binom{n-r}{k-t} \tag{17}$$

We set $\binom{n}{k} = 0$ whenever $k > n$ or $k < 0$. Equation (17) is known as the Vandermonde identity.

4.4.3 Binomial Theorem

Theorem 11 (Binomial Theorem)**.** For all $x, y \in \mathbb{R}$ and $n \in \mathbb{N}$ one has

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} \tag{18}$$

◁

Special choices of (x, y) in (18) give useful corollaries of Theorem 11:

If we set $x = y = 1$, we obtain the sum over one row in Pascal's triangle (18a). This is equal to the number of all subsets of an n-set and thus the size of the power set. Another interesting case is $x = -1$ and $y = 1$, which gives the alternating sum over one row in Pascal's triangle (18b).

$$x = y = 1 \Longrightarrow \sum_{k=0}^{n} \binom{n}{k} = 2^n \tag{18a}$$

$$x = -1, y = 1 \Longrightarrow \sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0 \tag{18b}$$

Remark 2 (Parity of bit strings)**.** Equation (18b) shows that, for every n, the number of even-parity n-bit strings (an even number of ones) equals the number of odd-parity strings. When n is odd this is witnessed by the involution that flips every bit, $w \mapsto \overline{w}$. For even n the equality follows algebraically from the vanishing alternating sum above. ◀

4.4.4 Approximations

Exact evaluation of $\binom{n}{k}$ can be expensive for large instances. Simple bounds follow directly from the factorial definition (Definition 37, Equation (2)). Rearranging the factors in (2) gives

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{\prod_{i=0}^{k-1}(n-i)}{\prod_{i=0}^{k-1}(k-i)} = \prod_{i=0}^{k-1} \frac{n-i}{k-i} \geq \prod_{i=0}^{k-1} \frac{n}{k} = \left(\frac{n}{k}\right)^k$$

How much bigger can $\binom{n}{k}$ be than $(n/k)^k$? Consider the following ratio:

$$\frac{\binom{n}{k}}{\left(\frac{n}{k}\right)^k} = \underbrace{\frac{\prod_{i=0}^{k-1}(n-i)}{\prod_{i=0}^{k-1} n}}_{\leq 1} \cdot \underbrace{\frac{k^k}{\prod_{i=0}^{k-1}(k-i)}}_{\leq e^k} \leq e^k$$

To upper bound the second part, we used the expansion of the exponential. One summand is less than the entire series.

Summarizing this we obtain the bounds:

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n}{k}\right)^k \cdot e^k \tag{19}$$

A sharper estimate uses Stirling's formula $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ to estimate the factorial and can thus be used to estimate the binomial coefficient:

$$\begin{aligned}\binom{n}{k} = \frac{n!}{k!(n-k)!} &\approx \frac{1}{\sqrt{2\pi}} \frac{\sqrt{n}}{\sqrt{k(n-k)}} \left(\frac{n}{e}\right)^n \left(\frac{e}{k}\right)^k \left(\frac{e}{n-k}\right)^{n-k} \\ &\approx \frac{n^n}{k^k (n-k)^{n-k}} = \frac{1}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} = \left(\frac{1}{\left(\frac{k}{n}\right)^{k/n} \left(\frac{n-k}{n}\right)^{(n-k)/n}}\right)^n\end{aligned}$$

In the last two steps we merely reformulate the expression in a form that turns out to yield some insight later.

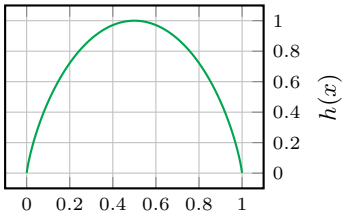
Now define $x := k/n$. Thus, $(n-k)/n = 1-x$, and we can write the estimate above as

$$\binom{n}{k} \approx \left(x^x(1-x)^{1-x}\right)^{-n} = 2^{n(-x\log_2 x-(1-x)\log_2(1-x))}$$

Further, we introduce the function

$$h(x) := -x\log_2 x - (1-x)\log_2(1-x)$$

The function h is symmetric about $x = \frac{1}{2}$, attains its maximum 1 there, and vanishes at the endpoints $x = 0$ and $x = 1$.



All this new formalism yields the estimate

$$\log_2 \binom{n}{k} \approx n \cdot h(x)$$

The function h plays an important role in information theory.

4.5 Special Counting Problems

4.5.1 Relations

A relation R on a set S is a subset of the Cartesian product $S \times S$ (Section 3.2, Definition 30). The number of relations on a set S of size n is given by:

$$\#R = |\mathbb{P}(S \times S)| = 2^{|S\times S|} = 2^{n^2} \tag{20}$$

because each element can either be in the relation or not (Ordered selection with repetition).

4.5.2 Equivalence Relations

Let S be an n-element set. We want to find the number of equivalence relations on S, i.e. the number of partitions of S into non-empty blocks (Bell number B_n). Instead of counting the equivalence relations directly, we first want to find the number of partitions of S into exactly k non-empty blocks (Stirling number of the second kind $S_{n,k}$). We can separate an element $s \in S$ and distinguish two cases:

1. s is in its own set and therefore adds a block by itself. Then we have to find a $k-1$ partition of the remaining $n-1$ elements.
2. s is in a set containing also other elements. Imaginge the remaining $n-1$ elements are partitioned into k sets. Then we could add s to any of those k sets.

This gives the recursion

$$S_{n,k} = S_{n-1,k-1} + k \cdot S_{n-1,k} \tag{21}$$

with base cases $S_{0,0} = 1$, $S_{n,0} = 0$, and $S_{n,n} = 1$. The total number of equivalence relations is thus

$$B_n := \sum_{k=0}^{n} S_{n,k} \tag{22}$$

which does not have a closed formula.

4.5.3 Permutations with Cycles

A permutation is a bijective map

$$\pi : \{1, \dots, n\} \to \{1, \dots, n\}$$

A common notation for permutations is

$$\pi = \begin{pmatrix} 1 & \cdots & n \\ \pi(1) & \cdots & \pi(n) \end{pmatrix}$$

A fixed point is an element i with $\pi(i) = i$.

Let S be an n-element set. We want to find the number of permutations of S with exactly k cycles (unsigned Stirling number of the first kind $S_{n,k}$). We can separate an element $s \in S$ and distinguish two cases:

1. s is a fixed point and therefore adds a cycle by itself. Then we have to find a permutation of the remaining $n-1$ elements with $k-1$ cycles.
2. s is in a cycle containing also other elements. Imaginge the remaining $n-1$ elements are permuted into k cycles. Then we could put s at $n-1$ positions, essentially after each of the existing elements (independent of which cycle they are in).

This gives the recursion

$$S_{n,k} = S_{n-1,k-1} + (n-1) \cdot S_{n-1,k} \tag{23}$$

with base cases $S_{0,0} = 1$, $S_{n,0} = 0$, and $S_{n,n} = 1$.

5 Graph Theory

5.1 Motivation

graphs are useful models for many discrete problems, because they allow to focus the analysis on specific relations between objects

5.2 Basic notions

Definition 38 (Graph)**.** A graph $G = (V, E)$ consists of

- a non-empty, finite vertex set V, i.e. $0 < |V| = n < \infty$.
- an edge-set $E \subseteq V \times V$: E is a relation on V. ✍

This definition allows for loops, but not for more than two edges connecting the same pair of nodes. Graphs not having this limitatation are called multigraphs.

Definition 39 (Undirected graph)**.** A graph G is undirected if

$$(u, v) \in E \Leftrightarrow (v, u) \in E$$ ✍

In this case, edges edges are sometimes taken to be unordered sets (instead of ordered pairs): $e = \{u, v\}$. When an undirected graph is drawn, a single line is normally drawn between u and v (instead of a pair of arrows).

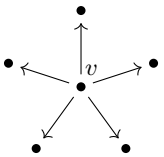
Definition 40 (Simple graph)**.** A graph G is called simple if it does not contain loops:

✍

Definition 41 (Neighbourhood)**.** For a vertex $v \in V$ we call the set

$$\Gamma(v) := \{w \in V \mid (v, w) \in E\}$$

the neighbourhood of v:



✍

Definition 42 (Degree)**.** The number of edges towards a node v is called the in-degree $\deg^-(v)$. The number of from a node v is called the out-degree $\deg^+(v)$. For undirected graphs the number of edges ending in a node v is simply called the degree $\deg(v)$. ✍

In a directed graph, any ingoing edge of a node is an outgoing edge of another node. Further, any edge is an ingoing and an outgoing edge for some node. Thus we obtain for directed graphs the following equation:

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

Lemma 12 (Handshaking)**.** Let $G = (V, E)$ be an undirected graph, where V is the vertex set and E the edge set. Then

$$\sum_{v \in V} \deg(v) = 2|E|$$ ◁

A consequence of Lemma 12 is that in every finite undirected graph, the number of vertices with odd degree is even.

Example 12 (Party)**.** If people shake hands at a party, the number of people who shake an odd number of other people's hands is even. ◀

5.2.1 Basic notions for simple undirected graphs

Definition 43 (Walk)**.** A v_1-v_l-walk with length l is a sequence of vertices $w = (v_1, \dots, v_l)$ with

$$(v_i, v_{i+1}) \in E \quad \forall i \in \{1, \dots, l-1\}$$ ✍

Definition 44 (Trail)**.** A v_1-v_l-trail is a v_1-v_l-walk for which all edges are distinct. ✍

Definition 45 (Path)**.** A v_1-v_l-path is a v_1-v_l-walk for which all vertices (and therefore also all edges) are distinct:

✍

Definition 46 (Circuit)**.** A circuit is a closed trail (Def 44), i.e. a sequence of vertices $c = (v_1, \dots, v_l)$ with

$$(v_i, v_{i+1}) \in E \quad \forall i \in \{1, \dots, l-1\} \quad \text{and} \quad (v_l, v_1) \in E$$ ✍

Definition 47 (Cycle)**.** A cycle is a closed path (Def 45), i.e. a circuit where all vertices are distinct. ✍

Definition 48 (Subgraph)**.** The pair $G' = (V', E')$ is a subgraph of $G = (V, E)$ if

$$V' \subseteq V \qquad E' \subseteq E \qquad E' \subseteq V' \times V'$$ ✍

Given a vertex set $V' \subseteq V$, the subgraph of G induced by V', denoted by $G[V']$, is the subgraph of G with vertex set V' and all possible edges which are also in G:

$$\forall u, v \in V' : (u, v) \in E \implies (u, v) \in E'$$

Definition 49 (Connected Components)**.** Let $(V_i)_{i \in I}$ be a partition of the vertex set of a graph G such that all elements within each partition (V_i) are connected by a path (Def 45). That is

$$\exists (u, v)\text{-path} \iff \exists i \in I : u, v \in V_i$$

Then the induced subgraphs $G[V_i]$ are called the connected components of G. ✍

Note that the existence of a path between vertices leads to an equivalence relation (Def 31) on the vertex set V. Correspondingly, it can also be seen as a partition of the vertex set V into equivalence classes.

Definition 50 (Bridge)**.** An edge $e \in E$ is called a bridge if the graph $G' := (V, E \backslash \{e\})$ has one more connected component than G. ✍

Theorem 13. A graph $G = (V, E)$ has at least $|V| - |E|$ connected components, i.e.

$$|V| - |E| \leq c$$

where c is the number of connected components. ◁

Proof. The graph $G = (V, \emptyset)$ has $|V|$ connected components, each containing one vertex. Any edge inserted into the graph reduced the number of connected components by at most one:

- If it connects two previously separate connected components, it reduces the number of connected components by one.
- If it connects two vertices within a connected component, it does not change the number of connected components. □

Corollary 14. If a graph $G = (V, E)$ is connected, i.e. $c = 1$, then

$$\boxed{|V| - |E| \leq 1}$$ ◁

In a minimally connected graph, the difference of the number of edges and the number of vertices reaches the bound from Corollary 14. All edges are bridges (Def 50). These minimally connected graphs are called trees (see Section 5.3).

5.3 Trees

A tree is a minimally connected graph. A graph containing trees as its connected components is called a forest.

Definition 51 (Forest)**.** An undirected (Def. 39), simple (Def. 40) graph without cycles (Def. 47) is a forest. ✍

Definition 52 (Tree)**.** A connected (Def. 49) forest is a tree. ✍

Definition 53 (Leaf)**.** A node $v \in V$ with $\deg(v) = 1$ is called a leaf. ✍

Theorem 15. A tree with at least two vertices has at least two leaves. ◁

Theorem 16 (Characterization of Trees)**.** Given an undirected graph $G = (V, E)$, the following statements are equivalent:

(1) G is a tree, i.e., a connected graph without cycles.

(2) G is connected and $|V| = |E| + 1$.

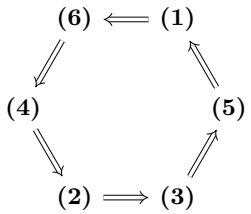
(3) G has no cycles and $|V| = |E| + 1$.

(4) G is connected and every edge is a bridge.

(5) G has no cycles and if an additional edge is added, it creates a cycle.

(6) For all vertices $u, v \in V$ there is a unique u-v-path. ◁

Proof. In total, $2 \cdot \binom{6}{2} = 30$ implications are claimed in Theorem 16. By transitivity of implication, it is sufficient to show a loop of implications. Any statement then follows from any other by following the implications in the loop. We can show the implications in the following cycle:



(1) ⇒ (6):

Let $u, v \in V$. Because G is connected (Definition 49), there is at least one u-v-path. It remains to show that this path is unique. Assume, for a contradiction, that there are two distinct u-v-paths. Starting at their first branching vertex, follow one path until they meet again, then return along the other path; this produces a cycle, contradicting **(1)**. Hence the u-v-path is unique, establishing **(6)**.

(6) ⇒ (4):

As there exists a path for any pair $u, v \in V$, we obtain immediately that G is connected. It remains to show that every edge is a bridge. Assume, for a contradiction, that there is an edge $e \in E$ that is not a bridge. Then, we can remove e and G is still connected, i.e., there is a path connecting the vertices adjacent to e. In other words, with e there are two paths connecting these two vertices: the original edge e and the detour through the remaining graph. This contradicts the uniqueness in **(6)**. Hence every edge is a bridge, proving **(4)**.

(4) ⇒ (2):

As connectedness is already given, it remains to show that $|V| = |E| + 1$. We show this by induction over the number of edges $|E|$.

(i) **Base case.** If $|E| = 1$, the edge joins two vertices, so $|V| = 2$. ✔

(ii) **Induction hypothesis.** Assume that every connected graph in which every edge is a bridge satisfies $|V| = |E| + 1$.

(iii) **Induction step.** As all edges are bridges, we can simply split the graph into two connected components by removing any of the edges: Then, we are left with two connected graphs. By (ii) and considering that all remaining edges are still bridges, we have

$$|E_1| + 1 = |V_1| \quad |E_2| + 1 = |V_2|$$

By adding these two equations, we obtain

$$\underbrace{|E_1| + |E_2|}_{=|E|-1} + 2 = |V_1| + |V_2| = |V|$$

and therefore $|V| = |E| + 1$. ✔

Thus **(2)** holds.

(2) ⇒ (3):

Let G be a connected graph with $|V| = |E| + 1$. Assume, for a contradiction, that G contains a cycle and let e be an edge of one such cycle. Since e connects two vertices u, v in the cycle, there is a path from u to v that does not use e. In other words, e is not a bridge (Def 50). Removing e therefore leaves the graph connected (since the rest of the cycle still links its endpoints), so $G' = (V', E') = (V, E \setminus \{e\})$ is connected with the same number of vertices but one edge less, i.e. $|V'| = |V|$ and $|E'| = |E| - 1$. For this new graph G' we have

$$|V'| - |E'| \overset{\lightning}{=} 2$$

which is impossible for a connected graph by Corollary 14. Hence no cycles exist and **(3)** follows.

(3) ⇒ (5):

We have to show that adding an edge e' to the graph G creates a cycle. First, we show by induction on $|V|$ that whenever $|E| \geq |V|$, there must be a cycle.

(i) **Base case.** The first interesting case is ($|V| = 3$). Three edges on three vertices force the triangle K_3, which has a cycle. ✔

(ii) **Induction hypothesis.** Assume a graph G with $|E| \geq |V|$ contains a cycle.

(iii) **Induction step.** We distinguish two cases:

If there exists a leaf in the graph, we merely remove that vertex and the corresponding edge. This reduces the number of vertices and the number of edges by one. If $|V| \leq |E|$ then also $|V| - 1 \leq |E| - 1$ and the graph contains a cycle by (ii). So, also the graph containing the leaf has a cycle. ✔

If there is no leaf, then all vertices have $\deg(v) \geq 2$. Then we can simply construct a cycle by going from one vertex to the next. As there is no leaf, there is no dead-end. As the number of vertices is finite, we have to end up at the initial vertex sooner or later and thereby closing a cycle. ✔

From **(3)** we have $|V| = |E| + 1 = |E'|$ for $E' := E \cup \{e'\}$. Thus $G' = (V, E')$ contains a cycle, as desired.

(5) ⇒ (1):

We already have that G is acyclic. So it remains to show that G is connected. Assume, for a contradiction, it were not connected, with two components G_1, G_2. Then, we could add a vertex e' to G by linking the two connected components. This yields a contradiction with **(5)**, as this insertion of an edge did not create a cycle. Hence G must be connected.

This closes the circle of implications, proving Theorem 16. □

5.3.1 Counting trees: Cayley's theorem

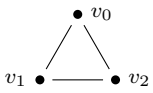
How many labeled trees exist with n vertices? The question can be rephrased, asking for the number of different spanning trees of the complete graph (clique) with n vertices, K_n.

Definition 54 (Spanning tree)**.** Given a connected graph $G = (V, E)$, a graph $H = (V, E')$ with the same vertex set is called a spanning tree of G if H is a tree (Def 52) and $E' \subseteq E$. ✍

Generally, there exist many different spanning trees for a given graph. An interesting algorithmic problem is to find the one that optimizes, for examle, the total weight, given that each edge has a weight. The most famous algorithms are greedy and due to Kruskal (1956) and Prim (1957).

Example 13. Let us first consider some cases for small n:

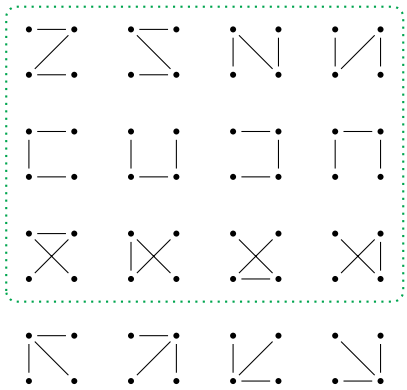
- $n = 1$: There is only one vertex and no edges and thus only one spanning tree.
- $n = 2$: There is merely one connected graph with two nodes. The single spanning tree is just the graph itself.
- $n = 3$: The completely connected graph K_3



has 3 spanning trees:

- $n = 4$: The completely connected graph K_4

has $12 + 4 = 16 = 4^2$ spanning trees:



We can group them into 2 sets: The "snake" type and the "star" type. The first 12, the "snakes", are the same if we consider them unlabeled: They are isomorphic (Def. 55).

- $n = 5$: There are 3 different shapes of trees with 5 vertices:
  - "snake": For any permutation of the 5 vertices, we obtain another isomorphic graph - except when we simply reverse the order. Therefore, there are $5!/2 = 60$ different labeled trees of this type.
  - "star": For each of the 5 vertices being in the center, we obtain a different graph. Swapping the leaves does not change the graph. So we have 5 different labeled trees of this type.

- "Y": This type has one vertex in the center, two short ends with 1 edge each (the upper part of the Y) and one long end with 2 edges (the lower part of the Y). The permutations of the vertices give the isometries. But swapping the two leaves at the short ends does not change the graph so we have again $5!/2 = 60$ different labeled trees of this type.

Thus, in total we have $60 + 5 + 60 = 125 = 5^3$ different labeled trees with 5 vertices. ◀

Definition 55 (Isomorphism)**.** Two graphs $G = (V, E)$ and $G' = (V', E')$ are called isomorphic, written $G \cong G'$, if there exists a bijection

$$\varphi : V \longrightarrow V' \quad \text{satisfying} \quad \forall\, u, v \in V : \ (u, v) \in E \iff \big(\varphi(u), \varphi(v)\big) \in E'$$

Such a function φ that preserves the edge relations is called an isomorphism. ✍

In particular, to count all labelled trees on n vertices one may first classify all unlabelled tree-shapes (up to isomorphism), and then multiply by the number of ways to assign labels to each shape (dividing out automorphisms of the shape). A more direct approach uses Cayley's theorem:

Theorem 17 (Cayley)**.** The number of different labelled trees on n vertices (equivalently, the number of spanning trees of the completely connected graph K_n) is

$$\boxed{n^{\,n-2}}$$ ◁

Proof. We construct a bijection

$$\big\{\text{trees on } [n] \text{ with two distinguished marks (head, tail)}\big\} \quad \longleftrightarrow \quad \big\{\, f\colon [n] \to [n] \,\big\}$$

Since there are n^2 ways to choose the head and the tail in a tree on n vertices, this implies that there are

$$\frac{n^n}{n^2} = n^{n-2}$$

different labelled trees on n vertices.

- function $\to$ tree
  (a) Start with any $f : [n] \to [n]$. Draw its functional digraph: each i points to $f(i)$. Every component has exactly one directed cycle.
  (b) Let $M \subseteq [n]$ be the set of all vertices on these cycles. On M, f restricts to a permutation.
  (c) Starting from the head, the i^{th} vertex on the spine is chosen as $f(M[i])$ where $M[i]$ is the i^{th} smallest element of M. So in particular the head is $f(M[1]) = f(\min M)$ and the tail is $f(M[|M|]) = f(\max M)$.
  (d) For every $v \notin M$ join v to $f(v)$. This yields an undirected, labeled tree with two additional marks: the head and the tail.
- tree $\to$ function
  (a) Given a tree on $[n]$ with a marked head and a marked tail, read off the unique simple path from head to tail; its vertices are exactly the set M.
  (b) Declare f to send $M[i]$ to the i^{th} vertex on the path from head to tail.
  (c) For any other vertex v, send $f(v)$ to the unique neighbor of v on the path toward the spine.

These two constructions are clear inverses of one another. Therefore the number of labeled trees with two marks is exactly the number of functions n^n, and dividing by the n^2 choices of the marks (head, tail) gives the claimed result. □

5.4 Some special graphs

Complete Graphs K_n (also called cliques) are graphs with n vertices an edge between every pair of vertices. For a complete graph K_n with n vertices, K_n, there are

$$|E| = \binom{n}{2} = \frac{n(n-1)}{2}$$

edges.

Cycles C_n are graphs with all vertices connected in a single cycle without additional edges. The smallest cycle is C_3 (triangle). The cycle C_n with n vertices has n edges.

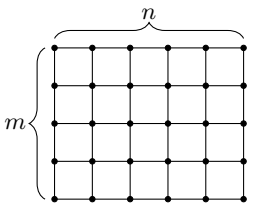
Mesh Graphs $M_{m,n}$ are graphs with a vertex set

$$V = \{(i,j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$$

and an edge set containing merely the closest neighbors of each vertex

$$E = \{((i_1,j_1),(i_2,j_2)) \mid |i_1 - i_2| + |j_1 - j_2| = 1\}$$

and no diagonals:



If we interpret the mesh in a cyclic way, we obtain different topological shapes, depending on how we connect the boundaries:

- Disk: If we do not connect any boundaries, we obtain a 2-dimensional orientable manifold with a single boundary component homeomorphic to a circle, and it embeds in $\mathbb{R}^2$ without self-intersection.
- Cylinder: The nodes are connected horizontally, i.e. left and right boundaries are connected, but the top and bottom are not:

  $$E_{\text{Cylinder}} = E \cup \{((i,1),(i,n)) \mid 1 \leq i \leq m\}$$

  It is a 2-dimensional orientable manifold with two boundary circles; although intrinsically flat, it requires embedding in $\mathbb{R}^3$ as a tube to avoid self-intersection.
- Möbius strip: The left and right boundaries are connected, but in reverse order:

  $$E_{\text{Möbius}} = E \cup \{((i,1),(m-i+1,n)) \mid 1 \leq i \leq m\}$$

  The Möbius strip is a 2-dimensional non-orientable surface with one boundary circle. Three spatial dimensions ($\mathbb{R}^3$) are needed to embed it without tearing or intersecting itself.
- Torus ('Donut'): The nodes are connected horizontally as well as vertically:

  $$E_{\text{Torus}} = \underbrace{E \cup \{((i,1),(i,n)) \mid 1 \leq i \leq m\}}_{E_{\text{Cylinder}}} \cup \{((1,j),(m,j)) \mid 1 \leq j \leq n\}$$

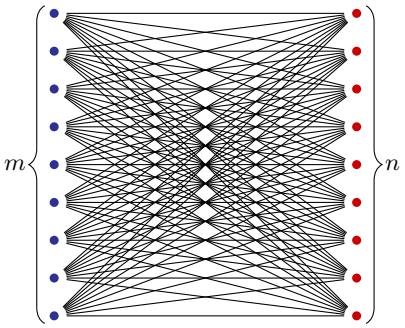
  It is a closed (no boundary), 2-dimensional orientable manifold of genus 1, and it embeds smoothly in $\mathbb{R}^3$ as the familiar donut shape.
- Klein bottle: The left and right boundaries are connected in reverse order (as in the Möbius strip), while the top and bottom boundaries are connected without reversal:

  $$E_{\text{Klein}} = \underbrace{E \cup \{((i,1),(m-i+1,n)) \mid 1 \leq i \leq m\}}_{E_{\text{Möbius}}} \cup \{((1,j),(m,j)) \mid 1 \leq j \leq n\}$$

  It is a closed (no boundary), 2-dimensional non-orientable manifold. Unlike a sphere or a torus, the Klein bottle does not have an inside or outside. Any realization in $\mathbb{R}^3$ self-intersects, so a true embedding requires $\mathbb{R}^4$.
- Projective plane: Both pairs of opposite boundaries are connected in reverse order:

  $$E_{\mathbb{RP}^2} = E \cup \{((i,1),(m-i+1,n)) \mid 1 \leq i \leq m\} \cup \{((1,j),(m,n-j+1)) \mid 1 \leq j \leq n\}$$

Complete bipartite graph $K_{m,n}$ is a graph with two sets of vertices V_1 and V_2 with $|V_1| = m$ and $|V_2| = n$, and an edge between every pair of vertices $v_i \in V_1$ and $v_j \in V_2$. It has $|V| = m + n$ vertices and $|E| = mn$ edges:



Hypercube Q_d is the d-dimensional hypercube with vertices that are d-bit strings, i.e.

$$V = \{0,1\}^d = \{(b_1, \ldots, b_d) \mid b_i \in \{0,1\}\}$$

It has $|V| = 2^d$ vertices. The edges are given by pairs with Hamming distance (Def 56) 1, i.e. the two strings differ in exactly one bit:

$$(u, v) \in E \quad :\Leftrightarrow \quad d_H(u, v) = 1$$

Every vertex has d neighbors, i.e., $\forall v \in V : \deg(v) = d$. The total number of edges is

$$|E| = \frac{d \cdot 2^d}{2} = d \cdot 2^{d-1}$$

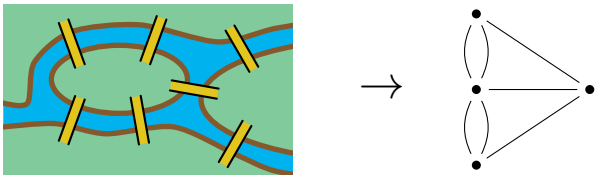
as each of the 2^d degrees is d. We draw the graph as follows: For obtaining the graph Q_{d+1} we draw twice the graph Q_d. To one of the two we add a 0 after all bit strings labelling the vertices, to the second respectively a 1. Finally, we add edges to connect the corresponding vertices of the two copies of Q_d. We start from Q_0, containing merely the empty word ϵ. Then the hypercube Q_1 is constructed following the procedure above. Repeating the same process yields Q_d for larger d.

Definition 56 (Hamming distance)**.** The Hamming distance between two bit strings of same length, $d_H(x, y)$, with $x, y \in \{0,1\}^d$ is the number of bits in which x and y differ. ✍

5.5 Euler Tours and Hamilton Cycles

In this section we will see two extrema of computational hardness (linear time vs. NP-complete) in two very similar-sounding problems.

Example 14 (Bridges of Königsberg)**.**



In 1736, Leonhard Euler considered the question of whether there is a tour crossing every of the seven bridges in the Prussian city of Königsberg exactly once. This question is often said to mark the beginning of graph theory. The situation is reflected in the above (multi-)graph. We are looking for a closed way that passes each edge exactly once - a so-called Euler tour (Def. 57). ◀

Definition 57 (Euler Tour)**.** An Euler tour is a closed sequence of edges of a graph $G = (V, E)$ that contains each edge of the graph exactly once. ✍

Theorem 18 (Euler)**.** A connected graph has an Euler tour if and only if all degrees are even. That is

$$G \text{ has an Euler tour} \quad \Leftrightarrow \quad \forall v \in V: \quad \deg(v) \text{ is even} \qquad \triangleleft$$

Proof. We first show that the condition 'deg(v) is even for all $v \in V$' is necessary for the existence of an Euler tour.

'$\Longrightarrow$':

This follows from the observation that any vertex is reached and left the same number of times. Whenever we reach or leave the edge, we have to use an other edge to form an Euler tour. Thus, the number of available edges has to be even if we finally have to have passed all edges. Note that the same holds for the starting vertex.

It remains to show that the condition is sufficient.

'$\Longleftarrow$':

We construct an Euler tour for a graph satisfying the condition above. First, we choose an initial vertex $v_1 \in V$. Following any yet unused edge one now proceeds to other vertices. As there is an even number of edges ending at each of the vertices, and as edges are always used (i.e., removed) in pairs (arrive + leave), we always finds such an unused edge for continuing, except at v_1. As the number of edges and the number of vertices are finite, we must eventually end up in v_1, and thus obtain a first closed way.

This way may not contain all edges. Then, as the graph is connected, there must be a vertex v_2 in the already-found way with still (at least two) unused edges. We use this vertex v_2 as the starting point of an additional way, using the same procedure of following the unused edges. Note that it is still true that all degrees in the graph are even since, again, we removed edges always in pairs with respect to any vertex: arrive and leave.

Repeatig this iteratively until we used all edges yields the desired Euler tour. □

As the degree of any of the vertices of the multigraph in Example 14 is 3 or 5, thus odd, there does not exist an Euler tour.

The complete graph K_n has an Euler tour if and only if n is odd:

$$\deg(v) = n - 1 \quad \forall v \in V$$

The hypercube Q_d has an Euler tour if and only if d is even:

$$\deg(v) = d \quad \forall v \in V$$

The proof of Theorem 18 uses a simple greedy[4] algorithm, that finds an Euler tour in $O(|E|)$ time. The given proof above uses a simple greedy algorithm, that finds an Euler tour in linear time in the number of edges $|E|$. Finding on Euler tour is, hence, among the computationally easiest problems (you simply have to go through the entire graph, and you have it). If we modify the problem a little replacing "edge" by "vertex", we end up with a hard problem, the Hamilton cycles.

Definition 58 (Hamilton Cycles)**.** A cycle that visits every vertex exactly once is called a Hamilton cycle. A graph containing such a cycle is called Hamiltonian. ✍

The cycles C_k are Hamiltonian for all $k \geq 3$.

The complete graphs K_n contain the cycles C_n for all $n \geq 3$, and therefore a Hamilton cycle. Thus they are Hamiltonian.

Wheel graphs W_n are Hamiltonian.

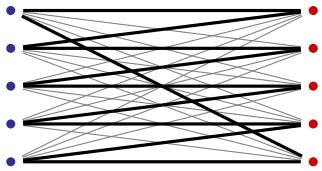
Trees are not Hamiltonian, as they contain no cycles at all.

Mesh graphs $M_{m,n}$ are Hamiltonian if and only if $m \cdot n$ is even. To see this, note that $M_{m,n}$ is bipartite. A bipartite graph can be Hamiltonian only if it contains the same number of nodes of each of the two colors, because in the (closed) Hamiltonian cycle, the

[4] At each vertex we choose any unused edge, independent of any previous choice. The choice does not depend on any global properties.

colors must switch in every step. Therefore the number of nodes ($m \cdot n$) must be even, showing that the latter is a necessary condition. If $m \cdot n$ is even, either m or n must be even. Without loss of generality, assume m is even (otherwise we can just rotate by 90°). Then we can construct a Hamiltonian cycle by starting at the top left corner and going down the first column, then going up in a zig-zag pattern through the remaining columns.

Complete bipartite graphs $K_{m,n}$ are are Hamiltonian if and only if $m = n$:



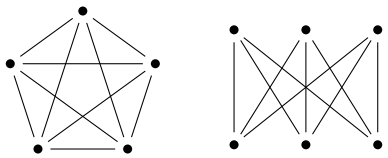
Hypercubes Q_d are Hamiltonian for $d \geq 2$. Q_0 and Q_1 are trees and thus cycleless. The square Q_2 is Hamiltonian (base case). Assume Q_d is Hamiltonian (induction hypothesis). To construct a Hamilton cycle in Q_{d+1} we remove two corresponding edges $(a_1 0, a_2 0)$ and $(a_1 1, a_2 1)$, and then add $(a_1 0, a_2 1)$ and $(a_1 0, a_2 1)$ (induction step).

The problem wether a general graph is Hamiltonian is believed to be hard: It is NP-complete.

5.6 Planar Graphs

Definition 59 (Planar graph)**.** A graph $G = (V, E)$ is planar if it can be drawn in the plane such that no edges cross. Such a drawing is called a planar embedding of G. Sometimes when we say planar graph, we refer to a planar embedding. ✍

Example 15. The complete graph K_5 and the complete bipartite graph $K_{3,3}$ are not planar:



◀

To actually see that the latter two graphs are not planar, we derive necessary properties all planar graphs share.

Definition 60 (Face)**.** A face or region f of a planar graph $G = (V, E)$ is a connected region of the plane. The set of regions is denoted by F. ✍

The boundary of a face is the subgraph formed from all vertices and edges that touch the face. Two faces are adjacent if their boundaries have at least one edge in common. A boundary walk of a face is a closed walk once around the perimeter of the face boundary. The degree of a face f is the length of the boundary walk of the face, denoted $\deg(f)$.

Lemma 19 (Faceshaking)**.** Let $G = (V, E)$ be a graph with a planar embedding, where F is the set of all faces. Then

$$\sum_{f \in F} \deg(f) = 2|E| \qquad \triangleleft$$

Proof. Each edge has 2 sides, and each side contributes 1 to a boundary walk, so the edge contributes 2 to the sum of degrees. □

Theorem 20 (Euler's polyhedron formula)**.** Let $G = (V, E)$ be a connected planar graph that divides the plane into $|F|$ regions (including the region outside the graph). Then

$$\boxed{|V| + |F| - |E| = 2}$$

But not the other way round! ◁

Proof. Theorem 20 is true for trees, as they have $|F| = 1$ and $|E| = |V| - 1$. We can now reduce any connected planar graph to a tree by removing edges from cycles. Removing an edge from a cycle reduces the number of regions and the number of edges by one, while the number of vertices remains the same. Repeating this process until there are no cycles left, we obtain a tree that satisfies Theorem 20 as seen above. Thus, Theorem 20 also holds for the initial planar graph. □

But if there are crossings, the notion of a region and thus also the number of regions $|F|$ is not defined. We find bounds on the number of regions $|F|$ in terms of the number of edges $|E|$.

Since a simple graph (Definition 40) does not contain loops or multiple edges between two vertices, every region must be surrounded by at least three edge-sides. The number of edge-sides is twice the number of edges, and thus

$$3|F| \leq 2|E| \implies |F| \leq \frac{2}{3}|E|$$

If the graph is bipartite, every region is bounded by at least four edge-sides, and thus

$$4|F| \leq 2|E| \implies |F| \leq \frac{1}{2}|E|$$

Inserting those formulae into Theorem 20 yields

Corollary 21. In a finite, connected, simple, planar graph $G = (V, E)$

$$|E| \leq 3|V| - 6 \tag{24}$$

if $|V| \geq 3$. If the graph is also bipartite, then

$$|E| \leq 2|V| - 4 \tag{25}$$

◁

Proof. If the graph is simple, then every face has at least 3 edges. Now $3|F|$ would count every edge 2 times, so we have $3|F| \leq 2|E|$. But $|E|+2 = |V|+|F| \leq |V|+2|E|/3$. So $3|E|+6 \leq 3|V|+2|E|$. So $|E| \leq 3|V|-6$. In a bipartite graph every face must have at least 4 sides. Thus $4|F| \leq 2|E|$, and the result follows similarly. □

Average degree The inequalities (24) and (25) imply for the average degree

$$\overline{\deg}(G) := \frac{1}{|V|}\sum_{v \in V} \deg(v) = \frac{2|E|}{|V|}$$

For a planar graph,

$$|E| \leq 3|V| - 6 < 3|V| \implies \overline{\deg}(G) < 6$$

For a bipartite planar graph,

$$|E| \leq 2|V| - 4 < 2|V| \implies \overline{\deg}(G) < 4$$

Characterizing planarity In Example 15, K_5 violates (24) and $K_{3,3}$ violates (25). In fact, it turns out that K_5 and $K_{3,3}$ are fundamental obstacles to planarity (Theorem 22).

A subdivision of a graph G is a graph G' obtained by inserting vertices into edges of G zero or more times.

Theorem 22 (Kuratowski)**.** A graph is planar if and only if it does not contain a subgraph that is a subdivision of K_5 or $K_{3,3}$ (Example 15). ◁

So every non-planar graph "contains" K_5 or $K_{3,3}$ in some sense.

5.7 Graph Colorings

Definition 61 (Coloring)**.** A k-coloring of a graph $G = (V, E)$ is a function

$$c : V \to \{c_1, \ldots, c_k\}$$

such that $c(v_1) \neq c(v_2)$ for all edges $(v_1, v_2) \in E$. ✍

Definition 62 (Chromatic number)**.** The chromatic number $\chi(G)$ of a graph $G = (V, E)$ is the minimal k such that a k-coloring of G exists. ✍

Example 16 (Chromatic numbers of special graphs)**.**

- Planar graphs G_p have $\chi(G_p) \leq 4$.
- Complete graphs K_n have $\chi(K_n) = n$ as every vertex needs its own color.
- The complete bipartite graphs have a chromatic number $\chi(K_{m,n}) = 2$ by definition.
- Mesh graphs are bipartite: $\chi(M_{m,n}) = 2$ (except for $m = n = 1$).
- Hypercubes are bipartite: $\chi(Q_d) = 2$ for $d \geq q$. This follows from an inductive argument: Obviously, Q_1 is two-colorable. If Q_d is twocolorable, then we can color the second Q_d with flipped colors before connecting the two to construct Q_{d+1}.
- For cycles, it depends on the parity of their length: $\chi(C_n) = \begin{cases} 2 & n \text{ even} \\ 3 & n \text{ odd} \end{cases}$
- Trees are two-colorable, $\chi(T) = 2$ (for any tree with at least an edge), as we can arrange the nodes in "levels". Then all levels with odd parity (depth) are colored in one color and those with even parity in another. ◀

Theorem 23. A graph $G = (V, E)$ is bipartite if and only if it does not contain an odd-length cycle. ◁

Proof. If G is bipartite, it cannot contain an odd cycle, since then $\chi(G) \geq 3$. It remains to show that, if G contains no odd cycle, then it is also bipartite. Let's consider a spanning tree of G. We can then color the levels with two colors. We now need to show that there are no edges in G that connect two levels of the same color. This follows from the observation that adding an edge connecting two vertices of levels with same parity always introduces an odd cycle, and thus a contradiction. □

Theorem 23 yields a linear-time algorithm to decide whether a graph is two-colorable. To decide whether a graph is 3-colorable is an NP-complete problem. Again, we see the two extremes of computational hardness in two very similar-sounding problems.