DISCRETE MATHEMATICS

ALEXANDER BORISOVICH

Swearword

Bastard Fr**e**d, who had f**u**n with his p**i**stol
Shooting **o**ne by one c**i**ty's old sp**i**nsters,
Was, with p**u**blic cond**e**mn,
Given **a**n AK**M**,
And all sp**i**nsters were g**o**ne in an **i**nstant.[1]

I have always considered the standard college course of Discrete Mathematics to be the only meaningful part of the lower-division math curriculum. After all, logic – sets – induction – recursion – modular arithmetic – combinatorics – probability – graph theory, however elementary, form a sapid chunk of real mathematics. So, when my son, aspiring to a EECS career, was allowed to skip this course in anticipation of a more advanced replacement, my parental pride was spiced with a grain of disappointment. Respectively, when (for reasons not worth explaining) he was nonetheless, on short notice, required to demonstrate his facility with these basics on the final exam of an ongoing course, I viewed it as a potentially useful challenge, especially since some help could come from me. Thus, I immediately located the detailed syllabus for the class online, and googled up the required textbook. To my surprise, in a few clicks, the free PDF file was downloaded on my laptop and *voila* — the entire text opened to my stupefied view: I was looking at a colorful Monster of one thousand letter-size pages long!

In its own way, it was a sweeping masterpiece. Indeed, what else is left to do in the business of ruining one's hope to understand mathematics — a subject which proudly strives for ultimate clarity, natural elegance, intellectual depth, economy of thought, and conciseness of representation — when the traditional math circle content that fits well in a middle-schooler's sandbox is watered down by a flood of biblical proportion to muddy the brain of a college student? Is there any

[1] Reverse-engineered from a notable Russian version of an unremarkable English limerick. In our age of prose poetry, the accents should help.

more cunning way to humiliate the Queen of Sciences, and compromise Her in the eyes of a sensible human being, than by ritualizing the "formal mathematical proof" and then practicing it for nauseatingly meticulous derivation of an infinitude of self-obvious trivialities? And then, when the greater half of the Monster is already behind, by stunning the reader with a new Principle, whose very name — *pigeonhole* — speaks of its accessibility to a 5th-grader, and "proving" it in a page-long argument sporting several displayed formulas?

As if to celebrate the success of this devilish enterprise, the records of the Monster's authors at the Math Genealogy Project show zero number of descendants, while their professional websites, shining with scores of educationists' publications, testify to the authors' integrity by showing zero math productivity of any kind.

To be fair, the course's professor contributed their bit by not following the Monster religiously, but rather omitting from the syllabus a few sections which had an unusually high concentration of genuine mathematical content (such as, e.g., the relation between path-counting in graphs and matrix multiplication), thereby saving time for, say, (apparently dear to their heart) fossilized tree-searching algorithms.

What the reader finds below is a result of my instinctive defensive reaction. With numerous swearwords and the aforementioned limerick (with no obvious connection at all) constantly reverberating in my mind, I dedicated a week of my sabbatical time to sifting the waters of the Monster for anything worthwhile, adding what had to be there but wasn't, learning what I never knew (e.g. the fossilized algorithms), correcting some crap found here and there, and ended up with what I propose as a sample of a new genre.

Doesn't a 1000-page book merit a 40-page book review? Here it is: *A comprehensive monster text review* — a concise, but comprehensive exposition of all the meaningful material to be included in a semester-long college course of Discrete Mathematics. It is equipped with a minimal but sufficient supply of examples and exercises, and with complete proofs, though occasionally presented as problem sets.

Read, solve, learn, enjoy — and shoot down all the monsters!

A COMPREHENSIVE MONSTER TEXT REVIEW

1. Logic and digital circuits

1. Operations with propositions. A *proposition* is a statement which is either TRUE (T) or FALSE (F), but not both. E.g. "x=7" and "The weather is good" are not propositions, but "2+2=5" is. We will denote propositions by the letters $p, q, \dots$ ("Boolean variables"). Here are some operations with propositions, resulting in new propositions:

- *negation* $\neg p$ TRUE whenever p is FALSE
- *conjunction* $p \wedge q$, TRUE whenever both p AND q are TRUE
- *disjunction* $p \vee q$, TRUE whenever one of p, q, or both are TRUE ("non-exclusive OR")
- *exclusive "or"* $p \oplus q$, TRUE whenever only one of p, q is (it is denoted so because if $T = 1$ and $F = 0$, then $p \oplus q$ is addition modulo 2)
- *conditional* (or *implication*) $p \to q$, FALSE only when the "hypothesis" p is TRUE, but the "conclusion" q is FALSE
- *bi-conditional* $p \leftrightarrow q$, which is TRUE whenever p and q have the same truth values

By the way, $p \to q \equiv \neg p \vee q$ ("either the hypothesis is FALSE, or the conclusion is TRUE"), where[2] the *equivalence* $\equiv$ of propositions means that both expressions are simultaneously TRUE and simultaneously FALSE for each possible combination of T/F values of the participating variables. Such identities between logical expressions in, say, n Boolean variables can be checked directly by "truth tables", i.e. by comparing the T/F values of the expressions for each of the 2^n combinations of the T/F values of the variables.

`Exercise:` Check the following equivalences by truth tables, or by using the previously checked ones (and correct one, which is wrong):

$$\neg(\neg p) \equiv p, \quad \neg(p \wedge q) \equiv \neg p \vee \neg q, \quad \neg(p \vee q) \equiv \neg p \wedge \neg q,$$

$$\neg(p \oplus q) \equiv \neg p \oplus \neg q, \qquad \neg(p \leftrightarrow q) \equiv p \oplus q, \qquad \neg p \equiv p \oplus T$$

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r), \quad p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

$$(p \to q) \not\equiv (q \to p), \; (p \to q) \equiv (\neg q \to \neg p), \; (q \to p) \equiv (\neg p \to \neg q)$$

The middle two in the top line are called *De Morgan rules*, while the third line contains *distributive* laws. Of course, $\vee$ and $\wedge$ are also associative and commutative.

[2] Note the convention on the "precedence order of operations": $\neg p \vee q \equiv r$ means $((\neg p) \vee q) \equiv r$ since $\equiv$, along with $\leftrightarrow$, are considered operations of the lowest order, $\to$, $\vee$, $\wedge$, and $\oplus$ of the next one, while $\neg$ of the highest.

The propositions in the bottom line illustrate "Alice in Wonderland": *direct* proposition $p \to q$ is not equivalent to the *converse* proposition $q \to p$, but is equivalent to the *contrapositive* proposition $\neg q \to \neg p$. The converse $q \to p$ is likewise equivalent to the *inverse* proposition $\neg p \to \neg q$. To prove a theorem $p \to q$ "by contradiction" means to prove $(p \wedge \neg q) \to F$ (assuming that the hypothesis is TRUE but the conclusion FALSE, arrive at a contradiction), while proving "by contraposition" means to prove the equivalent theorem $\neg q \to \neg p$ instead (assume that the conclusion is FALSE, and derive that the hypothesis must be FALSE too).

Boolean functions. There are 2^{2^n} *Boolean functions* $s(p_1, \dots, p_n)$ in n Boolean variables $p_1, \dots, p_n$ and taking Boolean values T/F: Each can be specified by the choice of a value T/F for each of the 2^n combinations of the values of the variables. For CS purposes, it is convenient to replace T and F with 1 and 0 respectively.

Every Boolean function can be expressed by using the operations $\neg, \vee, \wedge$. Indeed, for each string of input values $(p_1, \dots, p_n)$ (take for example $(1,1,0,1,0,0,1)$, where $n = 7$) write the unique conjunction monomial containing either p_i or $\neg p_i$ for each i which takes the value 1 on this particular input. In our example, it is

$$p_1 \wedge p_2 \wedge (\neg p_3) \wedge p_4 \wedge (\neg p_5) \wedge (\neg p_6) \wedge p_7.$$

By taking the disjunction ($\vee$) of all such monomials over those inputs for which our Boolean function takes the value $s = 1$ (and not including those for which it is 0) we obtain a Boolean expression representing our function. E.g. $s(0,0) = 0$ and $s(0,1) = s(1,0) = s(1,1) = 1$ is represented this way by $(\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2) \vee (p_1 \wedge p_2)$. Of course, this is not economical, because the function is $p_1 \vee p_2$.

There are various ways of producing minimal (in some sense, which can also vary) Boolean expressions for Boolean functions. As an example, consider *minimal disjunctive normal form*, which we won't define here formally. Rather we will give an example illustrating both what it is and how to construct one. One can think of the domain of a Boolean function as the set of vertices of the unit cube in n-dimensional space (because each vertex is specified by the string of n coordinates equal 0 or 1). For example, consider the Boolean function $s(p, q, r)$ equal to 1 at the vertices marked black in Figure 1, and equal to 0 at the "white" vertices. To produce a minimal disjunctive normal form, one needs to cover the set of all black vertices by as few "faces" of the cube (the "faces" can be of any dimension from 0 to n, and may overlap) as possible in such a way that they don't cover any white vertex. Thus, the 5 black vertices in the picture can be covered by two blue faces: one of dimension 2 (on top), the other of dimension 1 (the vertical blue line). The Boolean function, equal to 1 at the vertices of the top blue face only, is simply q. The Boolean function, equal to 1 at the vertices of the vertical blue edge, is $(\neg p) \wedge r$. Therefore $q \vee (\neg p \wedge r)$ (in general, the disjunction of the conjunction monomials representing the faces) is an economical Boolean expression for our function.

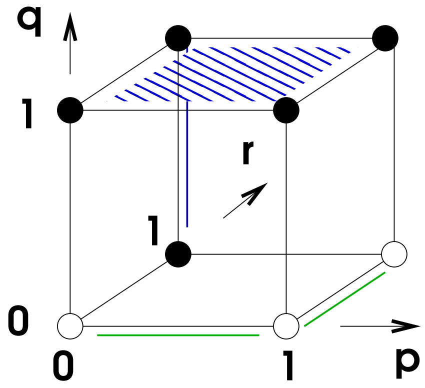

Figure 1

Note that the white vertices (defining the function $\neg s(p, q, r)$) can be covered by two green edges, leading to the expression

$$\neg s \equiv (\neg q \wedge \neg r) \vee (p \wedge \neg q)$$

and consequently

$$s \equiv \neg((\neg q \wedge \neg r) \vee (p \wedge \neg q)).$$

This representation of s is also "economical," but not disjunctive (because it is not a disjunction of conjunction monomials).

3. Digital logic circuits. In 1930s, C. Shannon noticed the analogy between electrical switchboards and Boolean functions. Namely, a Boolean expression involving $\wedge$ and $\vee$, but no negation, can be realized as such an electric circuit and vice versa. For example, the circuit shown in Figure 2, where each of the switches P, Q, R, S can be in either the *closed* position (TRUE) or the *open* position (FALSE), will light the bulb if and only if the Boolean expression $P \vee ((Q \vee R) \wedge S)$ assumes the value TRUE.

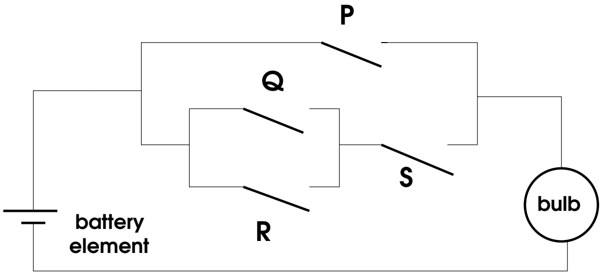

FIGURE 2

Modern logical circuits use a variety of technologies. Regardless of the technology, one can think of a circuit as a "black box" with n binary inputs $p_1, \dots, p_n$ and one binary output s, and encode the circuit's ability to transform the inputs into the output by the corresponding Boolean expression. Our problem of constructing economical expressions for Boolean functions translates into the task of realizing given "black boxes" in equivalent circuits using fewer NOT, AND, and OR *gates* (i.e. elementary circuits implementing these operations).

Thus, in our example in Figure 1, the two representations ("blue", and the negation of "green") of the function s correspond to the circuits shown in Figure 3.

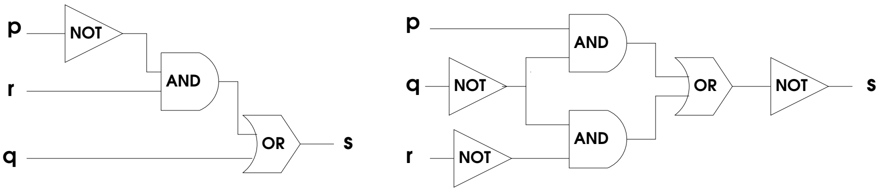

FIGURE 3

`Exercise.` Show that any logical circuit can be constructed using only two gates (Boolean operations): NAND, denoted by *Sheffer stroke* $|$, and NOR, denoted by *Pierce arrow* $\downarrow$, and defined by $p|q \equiv \neg(p \wedge q)$ and $p \downarrow q \equiv \neg(p \vee q)$ respectively. *Hint:* Combine $\neg p \equiv p|p$ with the De Morgan rules.

For example, the function $q \vee (\neg p \wedge r)$ from Figure 1 can be represented by the NAND/NOR circuit shown in Figure 4, i.e. as $((p|p) \downarrow q) \downarrow (q \downarrow r)$ (check this!)

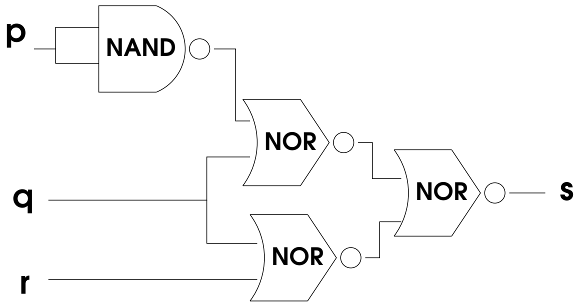

FIGURE 4

4. Circuits for binary addition and subtraction. Addition of two binary digits, p and q, can be expressed by *two* Boolean functions: the *sum* digit $p \oplus q$, i.e. $(p \vee q) \wedge \neg(p \wedge q)$, and the *carry* digit $p \wedge q$. The circuit in Figure 5 which realizes these functions is called a *half-adder*.

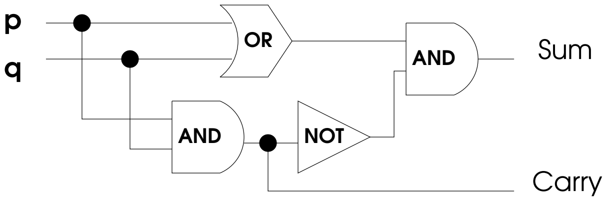

FIGURE 5. Half-adder

The circuit shown in Figure 6 is called a *full-adder*. It adds three binary digits, p, q, and r. In the binary addition of multi-digit numbers, the third summand may be a carry-over from the previous binary place. Check that it works! The key is that in the two additions needed for adding three binary digits, at most one carry-over can occur.

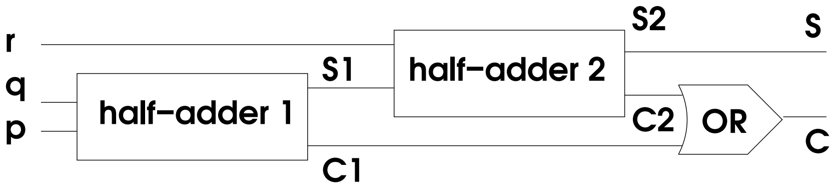

FIGURE 6. Full-adder

Now the addition of multi-digit binary numbers can be realized as shown in Figure 7, where $\dots p_2 p_1 p_0 + \dots q_2 q_1 q_0 = \dots s_3 s_2 s_1 s_0$.

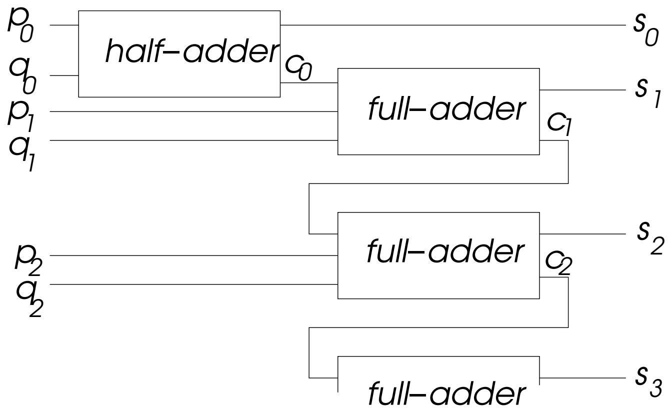

FIGURE 7. Multi-digit binary addition

5. Computer addition with negative numbers. One can use the n-digit binary format in order to represent integers a from the interval $-2^{n-1} \leqslant a < 2^{n-1}$. Namely, replace the integer a by its remainder $\widehat{a}$ modulo 2^n, which satisfies $0 \leqslant \widehat{a} < 2^n$. In practice this means that a non-negative $a < 2^{n-1}$ will be represented by 0 in the leftmost, nth place, followed by the $n-1$ digits of a, while negative $a \geqslant -2^{n-1}$ will be represented by $2^n - |a|$. It will have 1 in the nth place to signal that a was negative. Note that the binary code for $2^n - |a|$ is obtained by adding 1 to that of $2^n - 1 - |a|$, which is obtained from the code for $|a|$ by inverting all digits (think *why!*).

In the arithmetic modulo 2^n, we have $\widehat{a+b} = \widehat{a} + \widehat{b} \mod 2^n$. That is, assuming that the sum $a + b$ fits the range between -2^{n-1} and $2^{n-1} - 1$, the correct representation of $a + b$ is obtained by adding the representations $\widehat{a}$ and $\widehat{b}$ and, if the sum $\widehat{a} + \widehat{b}$ reaches 2^n, dropping the $(n+1)$st digit (i.e. reducing the sum modulo 2^n).[3]

6. Predicate logic. Suppose n denotes an integer, $n \in \mathbb{Z}$. Is $n > 5$ a proposition? No, because it is true for some n and is false for other n. If it is not a proposition, then what is it? The answer is that it is a function from $\mathbb{Z}$ to the set $\mathcal{P}$ of propositions: It is a rule, which to any given value of n (say, $n = 11$) associates a proposition ($11 > 5$ in this example), which is either true or false (true in this case). Such functions $p : X \to \mathcal{P}$ from sets to $\mathcal{P}$ are called *predicates*.

[3] Thus, mathematics relieves us from the need to consider the cases of various signs and sizes of a and b, as it is done in the Monster.

To each predicate, $p : X \to \mathcal{P}$ one can associate the *truth set* $T(p)$ which is a subset in X. By definition, it consists of those $x \in X$ for which the proposition $p(x) \in \mathcal{P}$ is TRUE. For example the truth set of the predicate $n > 5$ is the subset in $\mathbb{Z}$ of all integers greater than 5.

The *universal* ($\forall$) and *existential* ($\exists$) *quantifiers* are two operations, associating propositions to predicates. Namely, given a predicate $p : X \to \mathcal{P}$, we obtain propositions $\forall x\ p(x)$ and $\exists x\ p(x)$: the former is TRUE only when $T(p) = X$, i.e. when $p(x)$ is TRUE *for all* $x \in X$, and the latter is TRUE whenever $T(p) \neq \varnothing$, i.e. when *there exsts* $x \in X$ for which $p(x)$ is TRUE. Note that $T(p) = X$ is false whenever $T(\neg p) \neq \varnothing$, and $T(p) \neq \varnothing$ is false whenever $T(\neg p) = X$. Therefore

$$\neg(\forall x\ p(x)) \equiv \exists x\ (\neg p(x)), \quad \text{and} \quad \neg(\exists x\ p(x)) \equiv \forall x\ (\neg p(x)).$$

`Warning:` We often casually omit the logical quantifiers. For example, by writing the identity $p : a^2 - b^2 = (a+b)(a-b)$ we mean $\forall a, b \in \mathbb{R}, p(a,b)$. To accommodate this habit, let's use the notation $p(x) \Rightarrow q(x)$ to mean $\forall x\ (p(x) \to q(x))$.

`Exercises.` (a) *Disproving by counter-example:* Show that the theorem $p(x) \Rightarrow q(x)$ is FALSE whenever $\exists x\ (p(x) \wedge \neg q(x))$ is TRUE.

(b) Which of the implications $p(x) \Rightarrow q(x)$ and $q(x) \Rightarrow p(x)$ expresses that p is a *necessary* condition for q, and which that p is the *sufficient* condition for q?

(c) Show that $p(x) \Rightarrow q(x)$ is TRUE whenever $T(p) \subset T(q)$.

(d) Express contrapositive, inverse, and converse propositions, i.e. $\neg q(x) \Rightarrow \neg q(x)$, $q(x) \Rightarrow p(x)$, and $\neg p(x) \Rightarrow \neg q(x)$, as relationships between the truth sets $T(p)$ and $T(q)$.

(e) A real-valued function $f : \mathbb{R} \to \mathbb{R}$ of one real variable is said to be *uniformly continuous*, if

$$\forall \epsilon > 0\ \exists \delta > 0, \text{ such that } |x - y| < \delta \Rightarrow |f(x) - f(y)| < \epsilon.$$

Write down the negation of this proposition, and check whether the functions $f = x$ and $f = x^2$ are uniformly continuous.

(f) Change the order of the quantifiers in the above definition, i.e. write

$$\exists \delta > 0 \text{ such that } \forall \epsilon > 0, \text{ we have } |x - y| < \delta \Rightarrow |f(x) - f(y)| < \epsilon,$$

and describe all functions $f : \mathbb{R} \to \mathbb{R}$ which satisfy this condition.

2. Set theory

1. Operations with sets. By a *set* one means a collection of objects of any nature. The notion is so basic that it cannot be defined formally, simply because there are no more basic terms to rely on. The best one can do is to say that a set S is well-defined if for any object x in the Universe it is known whether x is an *element* of S ($x \in S$) or not ($x \notin S$). One says that T is a *subset* of S, $T \subset S$, if $x \in T \Rightarrow x \in S$. Two sets are called *equal*, $T = S$, if $T \subset S$ and $S \subset T$, i.e. if they consist of the same elements.

Warnings. (a) A set can be described by listing its elements, e.g. $S = \{2, 3, 5, 7\}$ is the set of all one-digit primes. However, lists with repeated objects, strictly speaking, don't describe sets, since elements of a set cannot be "repeated": they are either there or not.

(b) One can consider the set $\{3, \{3\}\}$, consisting of two elements: the integer $3 \in \mathbb{Z}$, and the subset $\{3\} \subset \mathbb{Z}$ consisting of one integer, 3. Yet, the statement $\{3\} \in \mathbb{Z}$ is false, since the elements of $\mathbb{Z} := \{0, \pm 1, \pm 2, \dots\}$ are numbers, not sets.

One defines the *complement* of a set, and the *intersection*, *union*, *difference*, and *symmetric difference* of two sets A and B by

$$\begin{aligned}(x \in A^c) &\Leftrightarrow (x \notin A)\\ x \in (A \cap B) &\Leftrightarrow (x \in A) \wedge (x \in B)\\ x \in (A \cup B) &\Leftrightarrow (x \in A) \vee (x \in B)\\ x \in (B - A) &\Leftrightarrow (x \in B) \wedge \neg(x \in A)\\ x \in (A \triangle B) &\Leftrightarrow (x \in A) \oplus (x \in B)\end{aligned}$$

Exercise. Prove the following properties of the operations:

$$\begin{aligned}(A^c)^c = A, \quad A \triangle B =& (A - B) \cup (B - A)\\ (A \cap B)^c = A^c \cap B^c, &\quad (A \cup B)^c = A^c \cap B^c\\ A \cap (B \cup C) =& (A \cap B) \cup (A \cap C)\\ A \cup (B \cap C) =& (A \cup B) \cap (A \cup C)\end{aligned}$$

Example: *Venn diagrams.* Let us prove a ridiculous identity about sets, e.g. that for any three sets A, B, C we have:

$$(A \cup B \cup C) - (A \triangle B) = (A \cap B) \cup (C - (A \cup B)).$$

We can use the definitions of the operations and translate this into a Boolean identity with three propositions — $a: \ x \in A$, $b: \ x \in B$, and $c: \ x \in C$:

$$(a \vee b \vee c) \wedge \neg(a \oplus b) \equiv (a \wedge b) \vee (c \wedge \neg(a \vee b)),$$

where the left (or right) side of $\equiv$ expresses the fact that x is an element of the set written on the left (resp. right) side of $=$. Now we can try to check the equivalence using truth tables. Consider, however, the example shown on Figure 8, where A, B, and C are three discs on the plane.

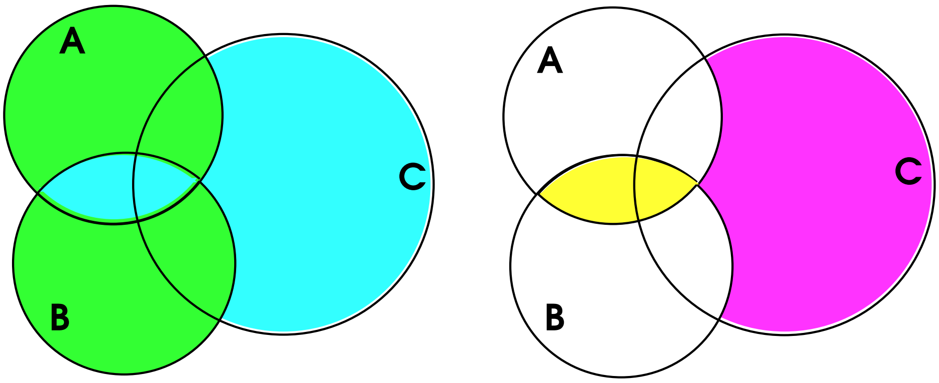

FIGURE 8. Venn diagrams

The area shaded green represents $A \triangle B$, and the complement of it inside $A \cup B \cup C$ (i.e. the *relative* complement) is shaded in cyan. In the right-hand picture, the relative complement of $A \cup B$ inside C is shaded in magenta, while the yellow region represents $A \cap B$. Since the total region shaded on the right coincides with the region shaded in cyan on the left, we conclude that at least for these three sets, our identity is true. So what? Have we proved the identity? Will it hold for any three sets? We claim that the answer is *yes*. The reason is that the three circles divide the plane into 8 regions (can you find all eight?) in which the predicates $x \in A$, $x \in B$, $x \in C$ assume all 8 possible truth values. Consequently, checking our identity for sets in this example is equivalent to verifying the Boolean identity by truth tables.

2. Boolean algebras. The parallelism between propositions and sets can be captured by the axiomatic algebraic structure called *Boolean algebra*. By definition, a Boolean algebra is a set $\mathcal{B}$ equipped with two operations, denoted $+$ and $\cdot$, which are required to satisfy the following properties (*axioms*):

- $+$ and $\cdot$ are commutative and associative, and obey two distributive laws: $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$ and $a + (b \cdot c) = (a+b) \cdot (a+c)$ for all $a, b, c \in \mathcal{B}$;
- there exist elements of $\mathcal{B}$ (denoted 0 and 1) such that $a + 0 = a$ and $a \cdot 1 = a$ for all $a \in \mathcal{B}$;

- for every $a \in \mathcal{B}$ there exists an element in $\mathcal{B}$ (denoted $\bar{a}$) such that $a + \bar{a} = 1$ and $a \cdot \bar{a} = 0$.

Taking $\mathcal{B}$ to be $\mathcal{P}$, the set of propositions, with $\vee$, $\wedge$, F, T, and $\neg$ in the roles of $+$, $\cdot$, 0, 1, and $^{-}$ respectively, we obtain the Boolean algebra of propositions.

Taking $\mathcal{B}$ to be the set of all subsets of a given set U, with $\cup$, $\cap$, $\varnothing$, U and $U - A$ in the roles of the respective operations, we obtain other examples of Boolean algebras.

From the axioms, many properties common to all Boolean algebras can be derived.

`Example:` Complements are unique. If $a + x = 1$ and $a \cdot x = 0$, then

$$\begin{aligned} x = x \cdot 1 = x \cdot (a + \bar{a}) &= x \cdot a + x \cdot \bar{a} = 0 + x \cdot \bar{a} \\ = a \cdot \bar{a} + x \cdot \bar{a} &= (a + x) \cdot \bar{a} = 1 \cdot \bar{a} = \bar{a}. \end{aligned}$$

`Exercise.` Derive the double-complement rule $\bar{\bar{a}} = a$, and the De Morgan rules: $\overline{a + b} = \bar{a} \cdot \bar{b}$ and $\overline{a \cdot b} = \bar{a} + \bar{b}$, to be true for all $a, b \in \mathcal{B}$.

3. Russell's paradox and Turing's halting theorem. It turns out that a naive approach to set theory leads to various contradictions. One of them, formulated by Bertrand Russell, shows than it is unsafe to use self-referential sentences to define sets. In a playful form it is usually formulated this way: "The barber in a town shaves everyone who doesn't shave himself; does the barber shave himself?" That is, does the barber belong to the set of all those he shaves, if the set is defined by his rule?

There is no answer: if he shaves himself, then he doesn't, and if he doesn't, then he does; i.e. the rule does not determine a set since it remains uncertain whether the barber himself is an element of this set or not.

An abstract version of this paradox can be illustrated by an attempt to define the set S of all those sets which do *not* contain themselves as their elements. Again, is S an element of S? If it is, then it isn't, and if it isn't, then it is. This contradiction shows that the phrase purporting to define S does not define any set.

While this (and some other) difficulties of set theory were resolved by a much more careful construction of its logical foundations, the paradoxes led to some fundamental "incompleteness" results in the work of Kurt Gödel and other logicians. Namely, mimicking the self-referential phrase *This statement is unprovable* (which, reasoning naively, is unprovable, because if it were provable, it would have been false, and false statements can't be proved, and so it is a true statement — yet unprovable!), Gödel managed to rigorously establish that not every *true* mathematical statement can be *derived* from the axioms of a given theory. Here is another similar development: The *halting theorem* by Alan Turing.

Theorem. *There is no algorithm, which for any algorithm X and any input data D would determine whether X with the input D will loop forever, or it will halt after finitely many steps.*

Proof. We will show that *any* algorithm, say $CheckHalts(X, D)$, which for every given algorithm X and every data D outputs either *halts* or *loopsforever*, will fail to function the required way on $X = Test$ and $D = Test$, where $Test$ is the following algorithm (considered as both an algorithm X and text input data D):

```
$Test(X)$;
  $a : IF\ \ CheckHalts(X, X) = halts$
      $THEN\ GO\ TO\ a$
      $ELSE\ GO\ TO\ b$;
  $b : STOP$
```

Indeed, when $CheckHalts(Test, Test) = halts$, $Test(Test)$ loops forever, and when $CheckHalts(Test, Test) = loopsforever$, $Test(Test)$ actually halts.

4. Functions. A *function* f from a set X (called the *domain* of f) to a set Y (called *codomain* of f) is a rule which to each element of X associates a unique element of Y. We write: $f : X \to Y$ for the function as a whole, but $X \ni x \mapsto f(x) \in Y$ to specify the element $y = f(x)$. We write $f(X)$ for the subset in Y called the *range* of f, and defined as $f(X) := \{y \in Y \mid \exists x \in X, \text{with } y = f(x)\}$.

A function is respectively called *injective*, *surjective*, or *bijective* if it is one-to-one (i.e. $(x_1 \neq x_2) \Rightarrow f(x_1) \neq f(x_2)$), onto (i.e. $F(X) = Y$), or both one-to-one and onto. One also says that a bijective function establishes a *one-to-one correspondence* between the sets X and Y: for every $y \in Y$ there exists (surjectivity) a unique (injectivity) $x \in X$ such that $y = f(x)$. This defines the function $f^{-1} : Y \to X$ called the *inverse* of f. As a mapping between sets, f^{-1} "undoes" what f does.

Example: *Hash functions.* How to efficiently store records of, say, up to $10,000$ students in a college, when the records are to be identified by 9-digit social security numbers (SSN)? A function $Hash$ from the set of 10^9 possible SSNs to the set of 10^4 labels ranging from 0000 to 9999 can be defined by a simple rule, e.g. by $Hash(SSN) :=$ the last 4 digits of the SSN. One problem is that this function might not be injective. When such a *collision* happens, a collision resolution algorithm is applied, e.g. one can assign the smallest allowed *Hash* value as yet unused by the database records. The resulting function *Hash* may look very irregular, but it will be injective.

Given two functions $g : X \to Y$, and $f : Y \to Z$, their *composition* $h := f \circ g : X \to Z$ is defined by $h(x) := f(g(x))$. The operation of composition of functions is associative: Given three functions, $h : X \to Y$, $g : Y \to Z$, and $f : Z \to W$, the composition of f with the result of composiing g and h coincides with the result of composing f and g, composed with h: $f \circ (g \circ h) = (f \circ g) \circ h : X \to W$. Indeed, both rules applied to $x \in X$ give the same function $f(g(h(x))$ (which is the only way of getting from X to W using f, g, and h). In fact, whenever there is an associative operation in mathematics, it can be defined in terms of the operation of composing functions. For example, the addition of plane vectors $\mathbf{u}+\mathbf{v}$ can be interpreted as the composition of *translations* on the plane: one by the vector $\mathbf{u}$, the other by the vector $\mathbf{v}$. Thus, the tautological associativity of function composition is the reason why this property is shared by many algebraic operations.

The *graph* of function $f : X \to Y$ is defined as the subset $F \subset X \times Y$ in the Cartesian product[4] of X and Y consisting of all pairs $(x, f(x))$, that is $F := \{(x, y) \in X \times Y \mid y = f(x)\}$. High-school students often fail to distinguish functions (which are rules, e.g. $x \mapsto x^2$) from their graphs, which are sets (parabola $y = x^2$ on the plane $\mathbb{R}^2 := \mathbb{R} \times \mathbb{R}$). In fact logicians do the same. Indeed, in order to reduce the number of undefinable notions, they define functions in terms of sets. Namely, a function from X to Y in their sense *is* a subset $F \subset X \times Y$ such that for every $x \in X$ there exists a unique $y \in Y$ such that $(x, y) \in F$. Of course, to arrive at this formal definition, one needs to have the informal idea of functions as rules beforehand.

`Exercises.` (a) Prove that $f : X \to Y$ is bijective (injective, surjective) if and only if f has a two-sided inverse (resp. left inverse, right inverse) with respect to the composition operation, i.e. if there exists $g : Y \to X$ such that $g \circ f = \mathrm{id}_X$ (a left inverse) and $f \circ g = \mathrm{id}_Y$ (a right inverse), where id_X and id_Y denote the identity maps on X and Y.

(b) Show that if $F \subset X \times Y$ is the graph of an invertible function, then the graph of the inverse function coincides with F considered as a subset of $Y \times X$ (which can be identified with $X \times Y$ by reordering the pairs).

[4] By definition, $X \times Y := \{(x, y) \mid x \in X, y \in Y\}$, i.e. the set of all ordered pairs.

5. Equivalence relations. Dealing with numbers, one often writes $x < y$ instead of simply saying: the ordered *pair* (x, y) is in the *binary* relation *"less than."* This example illustrates the informal idea of a binary relation. Speaking formally, the notion generalizes that of a function $X \to Y$ (identified with its graph). Namely, a *binary relation* between X and Y is any subset of their Cartesian product: $R \subset X \times Y$. One says that $x \in X$ and $y \in Y$ *are in the relation* R if $(x, y) \in R$. The statement that x and y are in this relation R can be expressed as xRy (like $x < y$), or by $R(x, y)$ (as in the Monster).

In mathematics, one often needs to partition a given set into non-overlapping subsets ("classes of equivalence" in a certain given sense), and introduce a new set, whose elements are the equivalence classes. For example, in arithmetic modulo 2 one deals with the set $\mathbb{Z}/2\mathbb{Z}$ of two equivalence classes into which the set of integers $\mathbb{Z}$ is partitioned according to their parity (even/odd). The idea of classification is formalized in the notion of equivalence relations. By definition, an *equivalence relation* on a set X is a binary relation $R \subset X \times X$ which is *reflexive*, *symmetric*, and *transitive*, i.e. respectively

$$\begin{aligned}
&\forall x \in X,\ R(x, x)\\
&\forall x, y \in X,\ R(x, y) \to R(y, x)\\
&\forall x, y, z \in X,\ R(x, y) \wedge R(y, z) \to R(x, z).
\end{aligned}$$

`Proposition - exercise.` For each $x \in X$, define the equivalence class $\widehat{x} := \{y \in X | R(x, y)\}$, and show that if R is an equivalence relation, then the equivalence classes form a *partition* of X, i.e. their union is the whole of X, and any two classes which have a common element coincide. Conversely, show that if the subsets $\widehat{x}$ form a partition of X, then R is an equivalence relation.

`Exercises.` (a) Which of the relations $x < y$, $x \leqslant y$, $x = y$ on $\mathbb{R}$ (meaning $X = Y = \mathbb{R}$, the set of reals) are reflexive? symmetric? transitive? Draw the corresponding subsets on the plane $\mathbb{R} \times \mathbb{R}$.

(b) Fix $n \in \mathbb{Z}$, and write $x \equiv y \mod n$ if[5] $n|(x - y)$. Check that the binary relation "$\equiv \mod n$" is reflexive, symmetric, and transitive, and show that the corresponding partition of $\mathbb{Z}$ is the partition according to the remainders of integers modulo n. What happens when $n = 0$?

Given an equivalence relation, one usually writes $x\neg y$ instead of $R(x, y)$. The set of equivalence classes is generally denoted by $X/\neg$, or X/R, though in the example from the previous exercise the standard notation is $\mathbb{Z}/n\mathbb{Z}$, or even shorter: $\mathbb{Z}_n$.

[5] The vertical bar | here means "divides".

6. Cardinality. How to find out whether two finite sets have the same number of elements? One way is to count their elements. In fact, counting is the process of establishing a one-to-one correspondence with some reference set. E.g. there are 5 business days in a week because I can establish a one-to-one correspondence between Mo, Tu, We, Th, Fr and the fingers of my right hand. Abstractly speaking, a counting number is the *class* of all those finite sets which can be put into one-to-one correspondence with each other. By the number of elements in a finite set we simply mean the class to which this set belongs.

The notion of cardinality extends this idea beyond finite sets. Two sets are said to *have the same cardinality* (written as $|X| = |Y|$, or $X \neg Y$), if there exists a one-to-one correspondence between them. One should check at this point that this is an equivalence relation. Therefore all sets are partitioned into equivalence classes according to their cardinalities.

Cardinalities of finite sets are usually denoted by non-negative integers $0, 1, 2, \dots, n, \dots$, where 0 stands for the class of the empty set $\varnothing$. These cardinalities (i.e. the classes of equivalence of finite sets) form a set denoted in the book by $\mathbb{N}$. It itself is infinite. Its subset $\{1, 2, 3, \dots\}$ (denoted in the book $\mathbb{Z}^+$) is also infinite, and seems smaller. However $|\mathbb{N}| = |\mathbb{Z}^+|$, since the function $n \mapsto n + 1$ from $\mathbb{N}$ to $\mathbb{Z}^+$ establishes a one-to-one correspondence.[6] Sets with the same cardinality as $\mathbb{Z}^+$ (or $\mathbb{N}$) are called *countable.*

`Exercises.` (a) Prove that the set $\mathbb{Z}$ of all integers (positive and negative) is countable.

(b) Prove that the union of countably many countable sets is countable. (*Hint:* Draw the 2-dimensional table whose rows contain the lists of elements of the given sets, and number all the items in the table by scanning it diagonally and skipping repeated items.)

(c) Prove that the set $\mathbb{Q}$ of all rational numbers is countable. (*Hint:* Write elements of $\mathbb{Q}$ as fractions m/n.)

(d) Prove that the real number line $\mathbb{R}$ has the same cardinality as any segment $(a, b) := \{x \in \mathbb{R} | a < x < b\}$ with $a < b$. (*Hint:* Use stereographic projection.)

(e) The same for (a, b) and $(a, b]$ or $[a, b]$, obtained from (a, b) by adding one of the endpoints or both. (*Hint:* Find a common countable subset in (a, b) and $(a, b]$.)

(f) Prove that the unit interval $[0, 1]$ and the unit square $[0, 1] \times [0, 1]$ have the same cardinality. (*Hint:* Encode a pair x, y of decimal

[6] By the way, the official definition of an *infinite* set is that its cardinality should not change when a new element is adjoined.

expansions $x = 0.x_1x_2\dots$ and $y = 0.y_1y_2\dots$, where x_i and y_i are decimal numerals, by a single expansion $z = 0.x_1y_1x_2y_2\dots$.)

Let us write $|X| \leqslant |Y|$ if X has the same cardinality as some subset of Y. The following celebrated *Cantor-Bernstein theorem* shows that such ordering of cardinalities is well-defined.

`Theorem.` *If* $|X| \leqslant |Y|$ *and* $|Y| \leqslant |X|$, *then* $|X| = |Y|$.

`Proof.` In math circles, the problem is playfully phrased in terms of two languages, Mumbo (M) and Jumbo (J), and two injective, but not surjective dictionaries: $f : M \to J$ and $g : J \to M$ (allowing each of the tribes to consider their language superior). The task is to construct a third dictionary, $h : M \to J$, which would be bijective. It has an elegant solution, which we recommend you to find on your own, but if you are lazy, here it is.

Since the functions are injective, inverse images are unique when they exist. So, starting with $m \in M$, we try to construct a sequence $m_0, j_0, m_1, j_1, \dots,$ where $m_0 = m$, $j_0 = g^{-1}(m_0)$ if it exists, $m_1 = f^{-1}(j_0)$, if it exists, and so on. Here are the possibilities: (i) the resulting sequence never terminates; (ii) it terminates in M: $m_0, \cdots, m_N$; (iii) it terminates in J: $m_0, \cdots, j_N$. We define $h(m)$ as $f(m)$ in the first two cases, and as j_0 in the third. The inverse function $h^{-1} : J \to M$ can be described this way: given $j \in J$, try to construct the sequence $j, m_0, j_0, \dots$ using inverse images as before. If it is infinite, then $m = m_0$ falls into type (i), and so $h^{-1}(j) = m_0$. If it is finite and terminates in M, then $m = m_0$ falls into type (ii), and so $h^{-1}(j) = m_0$ too. If it terminates in J, then $m = g(j)$ falls into type (iii), and so $h^{-1}(j) = g(j)$. (Indeed, the sequence built from m has the form $m, j, \dots, j_N$ (where possibly j is the last term), in which case $h(m) = j = g^{-1}(m)$.) Since both h and h^{-1} are well-defined everywhere on M and J respectively, they establish a one-to-one correspondence between M and J. Q.E.D.

7. Uncountability and non-computability. With the Cantor-Bernstein theorem at hands, it becomes obvious that the real line $\mathbb{R} = (-\infty, \infty)$, the half-line $(0, \infty)$, the segment $[0, 1]$, etc. have the same cardinality. It is called the cardinality of the *continuum*.

`Exercise.` Prove that the set, denoted $2^{\mathbb{N}}$, of all subsets in $\mathbb{N}$ (or in any other countable set) has the cardinality of the continuum. (*Hint:* Encode a subset by a string of binary digits which extends infinitely to the right, and fight the difficulties caused by the duplicate binary representation of some reals on $[0, 1]$, using the Cantor-Bernstein theorem.)

The following celebrated theorem by Georg Cantor is based on the *diagonal argument*, somewhat reminiscent of the self-referential algorithm in Turing's theorem.

`Theorem.` *The continuum is uncountable.*

`Proof.` We will show that any countable list of binary sequences cannot contain all such sequences. Indeed, if $s_1 = (s_1^1, s_1^2, s_1^3, \dots)$ is the first sequence in the list, $s_2 = (s_2^1, s_2^2, s_2^3, \dots)$ is the second, etc., where each $s_i^j = 0$ or 1, then the diagonal sequence $s = (s_1^1, s_2^2, s_3^3, \dots)$ shares with the sequence s_1 its first digit, with s_2 the 2nd, with s_3 the 3rd, and so on. Therefore the reverse sequence $\bar{s} = (\bar{s}_1^1, \bar{s}_2^2, \dots)$ (obtained from s by replacing 0s with 1s and 1s with 0s) is not in our list, since it differs from s_1 at its 1st place, from s_2 at the 2nd, from s_3 at the 3rd, and so on.

`Exercises.` (a) A real (or complex) number is called *algebraic*, if it is a root of a polynomial in one variable with integer coefficients. Prove the existence of non-algebraic numbers.

(b) Generalizing Cantor's theorem, prove for any set S that $|2^S| > |S|$, where 2^S denotes the set of all subsets in S (also denoted $\mathcal{P}(S)$ and called the *power set* of S). In other words, show that the cardinality of the power set is strictly greater than that of S, i.e. no injective function $f : S \to \mathcal{P}(S)$ is surjective. (*Hint:* Show that $\bar{F} := \{x \in \bar{F} \Leftrightarrow x \notin f(x)\}$ is not in the range of f.)

`Corollary` (existence of non-computable functions). *For every programming language, there exist functions $f : \mathbb{N} \to \{0, 1\}$ (or to any other non-trivial set) for which no program in this language computes f, i.e. for each input $n \in \mathbb{N}$ outputs $f(n)$.*

`Proof.` A program in a programming language is a text, i.e. a finite collection of symbols of a finite alphabet. All such sequences form a countable set. Indeed, there are finitely many such sequences of every fixed length, and so we can make a list of all sequences by first listing all one-letter sequences, then all two-letter sequence, and so on. Thus, there are countably many programs computing functions $\mathbb{N} \to \{0, 1\}$ (they form a subset in our countable list), but there are uncountably many such functions. Q.E.D.

3. Iterative processes

1. Induction. Consider an infinite sequence of propositions $p(n)$, $n = 0, 1, 2, \ldots$, and suppose that the following two theorems are proved:

$$\begin{array}{lll} 1^\circ & p(0) & \text{(the } \textit{base} \text{ of induction)} \\ 2^\circ & \forall n \in \mathbb{N}, p(n) \to p(n+1) & \text{(the } \textit{step} \text{ of induction).} \end{array} \cdot$$

The principle of *mathematical induction* guarantees that then all $p(n)$ are true: $1^\circ \wedge 2^\circ \Rightarrow \forall n, p(n)$. Indeed, to prove $p(n)$, start from $p(0) = T$, and conclude that $p(1) = T$ (since $p(0) \to p(1)$ is true), then conclude that $p(2) = T$ (since $p(1) \to p(2)$ is true), and *so on*, and arrive after n steps to $p(n) = T$.

A practical conclusion from this argument is that mathematical induction is a powerful logical tool (as it is indeed), because without it, in order to reach $p(1000)$, one would have to make a very long argument, repeating similar steps 1000 times. The drawback is that very often induction leads to a conclusive verdict, but the intuitive comprehension of *why* all $p(n)$ are true gets lost in the process. The theoretical aspect of this argument leads, however, to the notion of a *well-ordered set.*

A (partial) *ordering* on a set S is a binary relation R on it, which is reflexive ($\forall x \in S, xRx$), transitive ($(xRy) \wedge (yRz) \Rightarrow (xRz)$), and *anti-symmetric*: $(xRy) \wedge (yRx) \Rightarrow (y = x)$. Examples: (a) Divisibility $a|b$, on $\mathbb{Z}^+$. (b) Inclusion $\subset$, on the set $\mathcal{P}(U)$ of subsets in a given set. (c) Ordering of cardinalities $|X| \leqslant |Y|$ (thanks to Cantor–Bernstein).

A partial ordering is *total* if $\forall x, y \in S, (xRy) \vee (yRx)$. Examples: (a) $\leqslant$ on $\mathbb{R}$. (b) The ordering of words in a dictionary. (c) *Lexicographical ordering*, e.g. of points $(x, y) \leqslant (x', y')$ on the plane: first by $x \leqslant x'$, and if $x = x'$, then by $y \leqslant y'$ — as in the dictionary.

Suppose that every non-empty subset $X \subset S$ in a given totally ordered set has its minimal element with respect to R, i.e. there exists $x_0 \in X$ such that for all $x \in X$ we have $x_0 R x$. Then one says that S is *well-ordered* by R.

The truth is that $\mathbb{R}$ is not well-ordered by $\leqslant$ (because, e.g. the set of positive reals has no minimal element), but $\mathbb{N}$ is well-ordered by $\leqslant$: every non-empty subset of $\mathbb{N}$ has its minimal element.

The principle of mathematical induction can be proved on the basis of this property of $\mathbb{N}$. Namely, consider the set $X \subset \mathbb{N}$ of those n for which $p(n)$ is false. Assume $X \neq \varnothing$. Then X has the minimal element n_0. It cannot be 0, since $p(0) = T$ (the base). Therefore $n_0 - 1$ is defined, and $p(n_0 - 1) = T$. But since $p(n_0 - 1) \to p(n_0)$ is true (the step of induction), then $p(n_0) = T$, i.e. $n_0 \notin X$. This contradiction shows that $X = \varnothing$.

Here is an example illustrating why it is dangerous to discard the base of induction as “uninteresting”.

Problem: Find the number $R(n)$ of the regions into which n generic straight lines divide the plane. (“Generic” here means that no two are parallel, and no three are concurrent.)

Solution. Given $n-1$ lines on the plane, a generic nth line intersects each of the previous $n-1$ lines once. The $n-1$ intersection points divide the nth line into n parts. Therefore the nth line cuts through exactly n previous regions, dividing each of them in two, and creating therefore n new ones. Thus, $R(n) = R(n-1) + n$. Thus, $R(n) = 1+2+\cdots+n = n(n+1)/2$ is the nth *triangular* number. For example, $R(3) = 3(3+1)/2 = 6$ — Oops! $R(3) = 7$. The error comes from the fact that $R(0) = 1$, not 0, as for the triangular numbers, and so the right answer is $R(n) = 1 + (1+2+\cdots+n) = 1 + n(n+1)/2$.

The following elegant induction problem is probably included into every DM text. Removing one cell from a 2×2 chess-board, one obtains a shape called a *tromino*.

Problem: Show that any $2^n \times 2^n$ chess-board with one cell removed can be tiled by trominoes without overlaps.

Solution. Divide the board into four $2^{n-1} \times 2^{n-1}$ parts. The removed cell falls into one of them. Place one tromino at the center so that it covers one cell from each of the remaining three parts. Now it remains to tile the four parts, each with one cell removed. By the induction hypothesis (i.e. $p(n-1)$) this can be done, and $p(n)$ follows. Note that the base of induction ($p(1)$ in this case) is checked in the very formulation of the problem.

Exercises. (a) Use the solution of the tiling problem as an algorithm to tile the 8×8 chess-board with the cell $d7$ removed.
(b) Find another inductive solution, based on the partitioning of the board into $2^{n-1} \times 2^{n-1}$ cells of size 2×2 each.

Remark (on well-ordering). A countable set can be well-ordered (*why?*) Can this be generalized to sets of other cardinalities? It turns out (and this is a deep result of mathematical logic) that generally speaking, it is not possible to construct such a well-ordering, nor prove its existence in any other way, although assuming that it is possible would not lead to any contradiction. Thus, the *well-ordering principle*, saying that every set admits a well-ordering, can be accepted as an axiom.

2. Recursion. A *sequence* of elements of a set X is defined as a function $s: \mathbb{N} \to X$ (or $\mathbb{Z}^+ \to X$): $n \mapsto s_n$.

A *recursively defined* sequence is constructed by choosing $s_0 \in X$, and using the rule $s_n = f(s_{n-1})$ for each $n > 0$, where $f: X \to X$ is a given function.

`Example:` *2nd order linear recursion relations.* Such a relation has the form $a_{n+1} = \alpha a_n + \beta a_{n-1}$, where α and β are fixed numbers. Given a_0 and a_1, the rule determines a_n for all $n > 1$. In order to formally interpret it the rule as the recursively defined sequence in the above sense, one needs to take $X = \mathbb{R}^2$, $s_0 = (a_0, a_1)$, and define f as the linear map $f(x, y) := (y, \alpha y + \beta x)$. One can apply linear algebra in order to analyze the behavior of the dynamical system (with discrete time) in $\mathbb{R}^2$ defined by iterating the linear map f. Alternatively, note that if a'_n and a''_n are two sequences satisfying the given recursion relation, and γ' and γ'' are arbitrary numbers, then $a_n := \gamma' a'_n + \gamma'' a''_n$ also satisfies the recursion relation. (Check this linearity property!) This means that all solutions form a vector space. It has dimension 2, since a solution is determined by two constants: a_0 and a_1. So, let us look for particular solutions in the form $a_n = \lambda^n$. From the recursion relation, we find:

$$\lambda^{n+1} = \alpha\lambda^n + \beta\lambda^{n-1}, \quad \text{or, if } \lambda \neq 0\text{:} \quad \lambda^2 - \alpha\lambda - \beta = 0.$$

Let $\lambda_\pm$ be two distinct roots of this quadratic equation (called the *characteristic equation*). Therefore the general solution to this recursion relation has the form $a_n = \gamma_+\lambda_+^n + \gamma_-\lambda_-^n$ where $\gamma_\pm$ are arbitrary constants. In the notorious example of the Fibonacci sequence, we have: $a_{n+1} = a_n + a_{n-1}$, so that the characteristic equation is $\lambda^2 - \lambda - 1 = 0$, and has roots $\lambda_\pm = (1 \pm \sqrt{5})/2$. Thus, the general solution is

$$a_n = \gamma_+ \left(\frac{1+\sqrt{5}}{2}\right)^n + \gamma_- \left(\frac{1-\sqrt{5}}{2}\right)^n .$$

`Exercise.` (a) Find the coefficients $\gamma_\pm$ corresponding to the initial conditions $a_0 = 0$, $a_1 = 1$.

(b) Suppose the characteristic equation has the form $(\lambda - \lambda_0)^2 = 0$, i.e. has a double root λ_0. Show that the general solution to the corresponding recursion relation $a_{n+1} = 2\lambda_0 a_n - \lambda_0^2 a_{n-1}$ has the form $a_n = \gamma_0\lambda_0^n + \gamma_1 n\lambda_0^n$, where γ_0, γ_1 are arbitrary constants.

`Remark.` The idea of recursion goes beyond the notion of recursively defined sequences. For example, one can start with Boolean variables $p_1, \dots p_n$ and define Boolean expressions as obtained from them by recursively applying the operations $\vee, \wedge, \neg$. Or, one can recursively define vector spaces as linear subspaces in spaces of all vector-valued function $S \to V$, where S is any set, and V is any previously defined vector space, and equip the space (and its subspaces) with the operations of pointwise addition of functions and pointwise multiplication of functions by scalars. The proof of Turing's halting theorem was also based on the recursive use of algorithms $CheckHalts$ and $Test$. Iterative and self-referring processes are common in programming.

3. Integer division and correctness of algorithms. We know from first grade that division with remainders is recurring subtraction. Let us use a computer realization of the division algorithm to illustrate verification of algorithms. The division is realized by the code (in color) with commentaries in black:

```
$(0 \leqslant a \in \mathbb{Z}) \wedge (0 < d \in \mathbb{Z})$     precondition $Pre(x)$
$r := a;\ q := 0$                                              initial values $y := y_0$
while $(r \geqslant d)$                                         loop guard $G(x, y)$
$r := r - d;\ q := q + 1$                                      loop body $y := F(x, y)$
end while
% $(a = qd + r) \wedge (0 \leqslant r < d)$                   postcondition $Post(x, y)$
```

Thus, a general *while* loop receiving an input x and producing an output y consists of: the *precondition*, which is a predicate, $Pre(x)$, that has to be TRUE for the loop to start; the *initial condition* y_0 of the loop's output; the loop *guard*, which is a predicate, $G(x, y)$, that has to be FALSE for the loop to terminate; and the *body* of the loop, which iterates a function $y := F(x, y)$.

One says that the loop *functions correctly*, if it does terminate after finitely many iterations whenever $Pre(x)$ =TRUE, and produces the final output y such that the *postcondition*, which is a predicate $Post(x, y)$ (expressing the purpose of the loop), becomes TRUE.

`Proposition` (C.A.R. Hoare). *Let $I(x, y)$ be a* loop invariant*, i.e. a predicate satisfying:*

$$\begin{array}{ll} (i) & Pre(x) \Rightarrow I(x, y_0) \\ (ii) & G(x, y) \wedge I(x, y) \Rightarrow I(x, F(x, y)) \\ (iii) & \neg G(x, y) \wedge I(x, y) \Rightarrow Post(x, y). \end{array}$$

Then, assuming additionally that the loop guard is known to eventually becomes FALSE, the while loop functions correctly.

Indeed, (i) guarantees that the initial value of I is TRUE, (ii) guarantees that the value stays TRUE as long as the loop is iterated, and (iii) guarantees that when the loop stops, the output fulfills the purpose.

`Exercise.` Verify the division algorithm by checking that $I(a, d, r, q)$ defined as $0 \leqslant r = a - qd$ is a loop invariant.

4. Elementary number theory

1. The Euclidean algorithm. The following remarkable theory can be found (in a rather geometric form) in the oldest math textbook: *Euclid's "Elements"*. As we have just discussed, given two integers, a and $b \neq 0$, one divides a by b to obtain the *quotient* q and *remainder* r such that $a = qb + r$ and $0 \leqslant r < |b|$. A common divisor of q and r also divides $a = qb + r$, and a common divisor of a and b also divides $r = a - qb$. Therefore the set of common divisors of a and b coincides with the set of common divisors of b and r. The Euclidean algorithm proposes to replace the pair (a, b) with the pair (b, r), and continue the process: divide b by r with the remainder r_1 satisfying $0 \leqslant r_1 < r$, then divide r by r_1 with the remainder r_2 satisfying $0 \leqslant r_2 < r_1$, and so on. Of course, if a remainder turns out equal 0, the next division becomes impossible, and the process stops. Moreover, it is guaranteed to stop after $\leqslant |b|$ iterations, because the non-negative integer remainders decrease: $|b| > r > r_1 > r_2 > \cdots \geqslant 0$. Thus, we will have a finite succession of pairs

$$(a, b), (b, r), (r, r_1), (r_1, r_2), \ldots (d, 0),$$

(where we denoted the last non-zero remainder by d) with the property that they all have the same set of common divisors. Since the common divisors of d and 0 are the same as the divisors of d alone (which is, by the way, its own divisor), we arrive at the following conclusion:

`Euclid's theorem.` *For any integers a and $b \neq 0$ there is a unique positive integer d such that every common divisor of a and b is a divisor of d.*

Such d is called the *greatest common divisor* of a and b, and is denoted by $GCD(a, b)$ or simply by (a, b). E.g. $(a, b) = 1$ means that a and b have no non-trivial common divisors (i.e. none except ±1), in which case they are called *relatively prime.*

Moreover, writing out the steps of the Euclidean algorithm

$$\begin{array}{rlcrl} a = & qb + r & \Rightarrow & r = & a - qb \\ b = & q_1 r + r_1 & \Rightarrow & r_1 = & b - q_1 r \\ r = & q_2 r_1 + r_2 & \Rightarrow & r_2 = & r - q_2 r_1 \\ & & \ldots & & \\ r_{n-1} = & q_n r_n + d & \Rightarrow & d = & r_{n-1} - q_n r_n \\ r_n = & q_{n+1} d + 0 & & & \end{array},$$

we realize that each of the participating remainders is the sum of integer multiples of a and b. In particular, $d = ka + lb$ for some $k, l \in \mathbb{Z}$. This fact has important consequences.

Let us call an integer $p > 1$ *irreducible* if it cannot be factored in a non-trivial way (i.e. does not have divisors other than ± 1 and $\pm p$), and *prime*, if $p|ab \Rightarrow (p|a) \vee (p|b)$. Obviously, a prime p is irreducible, because existence of a factorization $p = ab$ with $|a|, |b| < p$ would contradict primality. Let us establish the converse: Assume that an irreducible p divides the product ab, but does not divide a, and deduce that p divides b. Indeed, since p has no other non-trivial factors than $\pm p$, and none of them divides a, we have $(a, p) = 1$ and hence (by the above observation about the Euclidean algorithm), $1 = ka + lp$ for some $k, l \in \mathbb{Z}$. Therefore $b = kab + lpb$, and since $p|ab$, we conclude that $p|b$ as promised.

Thus, irreducible positive integers $p = 2, 3, 5, 7, 11, 13, \ldots$ are the same as primes. One can derive by induction that if such p divides the product of several factors, then it divides at least one of them.

`Corollary` (the Fundamental Theorem of Arithmetic). *Any positive integer can be written as the product of powers of positive primes, and such a representation is unique up to reordering of the factors.* (Rephrasing: *Every* $n > 0$ *is uniquely written as* $n = 2^a 3^b 5^c 7^d 11^e \cdots$)

`Proof.` Existence of the factorization is almost obvious. If a given $n > 1$ is irreducible, then it is its own factorization. If it is reducible, factor it as $n = ab$, where $0 < a, b < n$, and continue factoring each of a and b if possible. Since the factors becomes smaller each time, the process eventually stops. yielding a factorization $n = p_1 \cdots p_N$ into primes. To prove uniqueness. assume that $p_1, \cdots p_N = q_1 \cdots q_M$, where all p_i and q_j are irreducible. Since p_1 divides the product $q_1 \cdots q_M$, it divides one of q_j (because an irreducible p_i is also prime!) Since q_j is irreducible, $q_j = p_1$. Thus, we can cancel p_1 with q_j, and proceed the same way with $p_2 \cdots p_N = q_1 \cdots \widehat{q_j} \cdots q_M$ (where the "hat" over q_j means that it is omitted). Eventually we will conclude that $N = M$, and that the list of q_j is obtained from the list of p_i by reordering.

`Exercise.` (a) Forget the previous theory, and start anew: Given $a, b \in \mathbb{Z}$, introduce[7] the set $I_{a,b} := \{ka + lb \parallel k, l \in \mathbb{Z}\} \subset \mathbb{Z}$, and prove that $I_{a,b}$ consists of all multiples of some $d \in \mathbb{Z}$. (*Hint:* Try the smallest positive number in $I_{a,b}$ in the role of d.)

(b) Show that $I_{a,b} = I_d := \{rd \parallel r \in \mathbb{Z}\}$ implies that $d = (a, b)$, i.e. that d is a common divisor of a and b, and that every other common divisor divides d.

(c) Prove that there are infinitely many primes. (*Hint:* None of a finite list of primes $p_1, \ldots, p_n$ divides $p_1 \cdots p_n + 1$.)

[7] We use $\parallel$ to avoid the collision with the "a divides b" notation $a|b$.

2. Modular arithmetic. We already know that for a fixed $n > 0$, the relation $a \equiv b \mod n$, meaning by definition that $n|(a-b)$, is an equivalence relation. It partitions $\mathbb{Z}$ into n equivalence classes (often called *congruence* classes of integers modulo n), uniquely represented by the remainders $0, 1, \ldots, n-1$.

Let us use the notation $\mathbb{Z}_n$ for the set of congruence classes modulo n. The operations of addition and multiplication of integers descend to $\mathbb{Z}_n$. Namely, define the sum and product of congruence classes $\widehat{a}, \widehat{b} \in \mathbb{Z}_n$ of two integers $a, b \in \mathbb{Z}$ as the congruence classes of their sum and of their the product respectively:

$$\widehat{a} + \widehat{b} := \widehat{a+b} \text{ and } \widehat{a}\widehat{b} := \widehat{ab}.$$

Exercises. (a) Check that the result does not depend on the choice of class representatives, i.e. that ($a' \equiv a \mod n$ and $b' \equiv b \mod n$ imply $(a'+b') \equiv (a+b) \mod n$ and $a'b' \equiv ab \mod n$. Thus, the map $\mathbb{Z} \to \mathbb{Z}_n : \ x \mapsto \widehat{x}$ transforms sums to sums and products to products.

(b) Show that when $n|N$, there is a unique map $\mathbb{Z}_N \to \mathbb{Z}_n$ which transforms sums to sums and products to products. (*Hint:* $a \equiv b \mod N$ implies $a \equiv b \mod n$.)

Theorem (The Chinese Remainder Theorem). *If* $(m,n) = 1$ *then the map* (from the previous exercise) $\mathbb{Z}_{mn} \to \mathbb{Z}_m \times \mathbb{Z}_n$ *is bijective.*

Proof. The map takes the congruence class of an integer a modulo mn into the pair of congruence classes of a — one modulo m, the other modulo n.[8] The total number of such pairs is mn, the same as the cardinality of $\mathbb{Z}_{mn}$. So, it suffices to prove that the map is injective, i.e. to show that if $a \equiv b \mod m$ and $a \equiv b \mod n$, then $a \equiv b \mod mn$. Indeed, for $c = a - b$, we are given that $m|c$ and $n|c$. Since m and n have no common factors, the Fundamental Theorem of Arithmetic implies that c must be divisible by their product. Q.E.D.

Exercise. Show that the map $\mathbb{Z}_{mn} \to \mathbb{Z}_m \times \mathbb{Z}_n$ is *not* injective whenever $GCD(m,n) > 1$. (*Hint:* $LCM(m,n) < mn$.)

In $\mathbb{Z}$, only ± 1 are multiplicatively invertible. Our goal is to study the congruence classes in $\mathbb{Z}_n$ which are invertible with respect to multiplication. We claim that $\widehat{a} \in \mathbb{Z}_n$ is invertible if and only if $(a,n) = 1$.

Indeed, if they have a common factor, i.e. $a = a'd$, $n = dn'$ for some $d > 1$, then $\widehat{a}\widehat{n}' = \widehat{a'dn'} = \widehat{a'n} = \widehat{0}$ in $\mathbb{Z}_n$. This implies that $\widehat{a}$ is not invertible, since if it were, we would multiply $\widehat{a}\widehat{n'}$ by the inverse of $\widehat{a}$ and conclude that $\widehat{n}' = 0$, which is not true.

[8] Note that it maps sums and products to (componentwise) sums and products, since according to the previous exercise, each of $\mathbb{Z}_{mn} \to \mathbb{Z}_m$ and $\mathbb{Z}_{mn} \to \mathbb{Z}_n$ does.

Conversely, when $(a,n)=1$, the Euclidean algorithm gives us k and l such that $ka+ln=1$. Reducing this modulo n we find $\widehat{k}\widehat{a}=1$, i.e. $\widehat{k}$ is the multiplicative inverse of $\widehat{a}$.

`Exercises.` (a) Show that multiplicatively invertible congruence classes (called *units* of $\mathbb{Z}_n$) form a subset (denoted $\mathbb{Z}_n^\times$) which is closed with respect to multiplication.

(b) Find all units in $\mathbb{Z}_8$ and analyze their multiplication table. Do the same for $\mathbb{Z}_5$.

(c) Prove that $|\mathbb{Z}_{p^k}^\times| = p^k - p^{k-1}$, where p is prime and $k>0$. (*Hint:* Which of numbers $1,2,\dots,p^k$ are not relatively prime to p^k?)

(d) Prove that for m,n relatively prime, the number of units modulo mn is the product of the numbers of units modulo m and n separately. (*Hint:* Show that $\mathbb{Z}_{mn}\to\mathbb{Z}_m\times\mathbb{Z}_n$ maps $\mathbb{Z}_{mn}^\times$ bijectively to $\mathbb{Z}_m^\times\times\mathbb{Z}_n^\times$.)

`Remark.` The *Euler function* φ associates to an integer $n>1$ the number of integers between 1 and n which are relatively prime to n. In our notation, $\varphi(n) := |\mathbb{Z}_n^\times|$, the number of units in $\mathbb{Z}_n$. The previous exercises imply that in terms of the prime factorization $n=p_1^{k_1}\cdots p_r^{k_r}$, where $p_1,\dots,p_r$ are *distinct* prime divisors of n, we have

$$\varphi(n) = p_1^{k_1-1}\cdots p_r^{k_r-1}(p_1-1)\cdots(p_r-1) = n\times\left(1-\frac{1}{p_1}\right)\cdots\left(1-\frac{1}{p_r}\right).$$

Indeed, the factors $p_i^{k_i}$ corresponding to different primes are pairwise relatively prime. Thus, applying Exercise (d) inductively, we find that $\varphi(n)=\varphi(p_1^{k_1})\cdots\varphi(p_r^{k_r})$, while Exercise (c) determines the value of each factor: $\varphi(p^k)=p^k-p^{k-1}=p^{k-1}(p-1)$.

`Fermat's Little Theorem.` *For all $a\in\mathbb{Z}$ and a prime p, we have $a^p\equiv a \mod p$.*

`Proof.` In other words, $\widehat{a}^p=\widehat{a}$ in $\mathbb{Z}_p$. Since this is obvious for $\widehat{a}=0$, and all other congruence classes in $\mathbb{Z}_p$ are invertible (because p is prime), it suffices to prove that $\widehat{a}^{p-1}=\widehat{1}$ for all $\widehat{a}\in\mathbb{Z}_p^\times$.

Take the string $\widehat{1},\widehat{2},\dots,\widehat{p-1}$ and multiply each term by $\widehat{a}$. This will result in the string $\widehat{1}\widehat{a},\widehat{2}\widehat{a},\dots,\widehat{(p-1)}\widehat{a}$, which is in fact just a permutation of the original string, because the operation of multiplication by $\widehat{a}$ is invertible. Multiplying the terms of the string, we conclude that $\widehat{(p-1)!}\,\widehat{a}^{p-1}$ coincides with $\widehat{(p-1)!}$ Since the factorial product is invertible in $\mathbb{Z}_p^\times$, we conclude that $\widehat{a}^{p-1}=\widehat{1}$ as required. Q.E.D.

`Exercises.` (a) Use the same reasoning to prove *Euler's Theorem*: If $(a,n)=1$, then $a^{\varphi(n)}\equiv 1 \mod n$.

(b) Compute $2^{2017} \mod 111$.

3. Public key cryptography. It is an elegant idea of a number-theoretic origin which enables one to receive messages from anyone in the world, encrypted by a publicly announced cipher, and yet decipherable only by the recipient, so that no third party can eavesdrop on the content of the messages. There exist some relatively efficient algorithms which allow one to determine whether some large numbers (say, consisting of hundreds of decimal digits) are prime or composite (without actually factoring them in the latter case), though factoring the product pq of two such prime numbers p and q would be unfeasible for modern computers in any reasonable time.

So, the recipient, being in possession of two such primes, publicly announces their product, R, and one more number, e, which must be relatively prime to $\varphi(pq) = (p-1)(q-1)$. Messages to the recipient must be expressed in integers and partitioned into "packets" of size $< R$. To send such a "packet" M, the sender transmits the remainder $C \equiv M^e \mod R$. A potential eavesdropper can see C, which will have the form $M^e - xR$ for some x, and though R and e are publicly known, the uncertainty in the number x of "overflows" would prevent the eavesdropper from recovering the message M.

Now, with probability close to 1 (compute it!) the number M is relatively prime to R, and hence, according to Euler's theorem, $M^{\varphi(R)} \equiv 1 \mod R$. Since e is relatively prime to $\varphi(R) = \varphi(pq) = (p-1)(q-1)$, it is invertible in $\mathbb{Z}_R$. The recipient can pre-calculate d such that $ed \equiv 1 \mod \varphi(R)$. Thus, the deciphering procedure consists in computing the remainder of C^d modulo R: $C^d = M^{ed} \equiv M \mod R$. Since the initial packet M is smaller than R, the remainder coincides with M.

In order to determine d from e and R, one needs to know $\varphi(R)$. Even if it is known that R is the product of two primes, p and q, determining $\varphi(pq) = (p-1)(q-1) = R + 1 - (p+q)$ is equivalent to finding the sum $p + q$, which together with the product $R = pq$ would determine p and q. Thus, for anyone who cannot find the factorization of R, the message C would remain undecipherable.

`Exercise.` Take small numbers for the public key: $R = 55$, $e = 3$.
(a) Encrypt message $M = 8$. Answer: $C = \dots$
(b) Determine the inverse of e modulo $\varphi(55)$. Answer: $d = \dots$
(c) Decipher the encrypted message: $C^d \equiv \dots \mod 55$.

`Example.` How to raise a to the power 27 modulo 55? Write $27_{10} = 11011_2$ (in binary code). Therefore $a^{27} = a^{16} \times a^8 \times a^2 \times a^1$. E.g., to compute $17^{27} \mod 55$, we find $17^2 \equiv 14 \mod 55$, $\mathbf{17}^4 \equiv 14^2 \equiv -24 \mod 55$, $17^8 \equiv (-24)^2 \equiv 26 \mod 55$, $17^{16} \equiv 26^2 \equiv 16 \mod 55$. Hence $17^{27} = 17^{16} 17^8 17^2 17^1 \equiv 16 \cdot 26 \cdot 14 \cdot 17 = 99008 \equiv 8 \mod 55$.

5. Combinatorics

1. The Pigeonhole Principle says that if $n+1$ items are placed into n compartments, then at least one of the compartments contains more than one item. Scientifically speaking, this means that no function $f : X \to Y$ is injective if $|Y| < |X| < \infty$. A proof, if desired at all, can be done by induction on the cardinality $n = |Y|$. The principle is clear when $Y = \varnothing$ (for there are no functions from $X \neq \varnothing$ to $\varnothing$), and for $n > 0$, if $|f^{-1}(y_0)| \leqslant 1$, the problem is reduced to the case $n-1$ for the function from $(X - f^{-1}(y_0)) \to (Y - \{y_0\})$.

`Exercises`. (a) Show that in any group of people, there are at least two who have the same number of friends within the group.

(b) Prove that the decimal (or binary) expansion of any fraction m/n repeats (starting from some place) and has a period fewer than n places. (*Hint:* The process of long division of m by n (let $n > m$) retraces itself as soon as the remainder repeats.)

2. Permutations, arrangements, combinations. Invertible functions from a set to itself are called *permutations*. When the set is finite (e.g. $S = \{1, 2, \ldots n\}$), the total number of such permutations equals $n! := n \cdot (n-1)! = n \cdot ((n-1) \cdots 2 \cdot 1)$. Indeed, after making one of the n choices for $f(1)$, there remains $n-1$ choices for $f(2)$, after which, there remains $n-2$ choices for $f(3)$, etc.[9]

`Exercises.` (a) Show that the number $A_{n,r}$ of ordered r-tuples of elements of $\{1, \ldots, n\}$ (they are called *arrangements*, or *r-permutations*) is equal to $n!/(n-r)! = n \cdot (n-1) \cdots (n-r+1)$ (assuming that $0 \leqslant r \leqslant n$, of course).

(b) Show that the number, denoted $C_{n,r}$ or $\binom{n}{r}$ ("n choose r"), of *un*ordered r-tuples of elements from $\{1, \ldots, n\}$ (they are called *r-combinations*, but are, simply speaking, r-element subsets) is equal to $n!/(n-r)!r!$

(c) Prove that for prime p and $0 < r < p$, the integer $p!/r!(p-r)!$ is divisible by p.

(d) Find how many times a prime p occurs in the prime factorization of $n!$ (*Answer:* $\lfloor n/p \rfloor + \lfloor n/p^2 \rfloor + \cdots$ where $\lfloor x \rfloor$ is the *integer part*, or *floor function*, assigning to $x \in \mathbb{R}$ the greatest integer not exceeding x.)

(e) An expression $x_1^{r_1} \cdots x_n^{r_n}$, where each r_i is a non-negative integer, is called a *monomial* in n variables $(x_1, \ldots, x_n)$ of total *degree* $r = r_1 + \cdots + r_n$. Prove that the total number of monomials in n variables

[9] One can organize this argument as formal induction by showing that the number of permutations, $p(n)$, satisfies $p(n) = n \cdot p(n-1)$ for all $n > 0$. Can you explain, though, why $p(0) = 1$ (the base of induction)?

of total degree r is equal to $\binom{n+r-1}{r}$. (*Hint:* Placing $r-1$ crosses "×" in the string of $n+r-1$ blank spaces divides the spaces remaining blank into r, possibly empty, groups.)

(f) An (unordered) collection of possibly repeated letters taken from a certain alphabet is called a *multiset*. How many multisets of size r can be made of n letters? What if the order of the letters also matters?

3. The binomial formula. Multiplying out n copies of the factor $(x+y)$, we find

$$\begin{aligned}(x+y)^n =&(x+y)(x+y)\cdots(x+y)\\ =&\binom{n}{0}x^ny^0+\binom{n}{1}x^{n-1}y+\cdots+\binom{n}{n}x^0y^n=\sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k.\end{aligned}$$

Indeed, the number of times a monomial $x^{n-k}y^k$ occurs in the sum equals the number of ways to choose k letters y from among the n parentheses $(x+y)$, i.e. "n choose k" times.

We already know how to express the values of the *binomial coefficients* using factorials: $\binom{n}{k}=\frac{n!}{k!(n-k)!}$. It turns out however, that the properties of these numbers are more clearly encoded in the properties of the expression $(x+y)^n$, sometimes referred to as the *generating function* for the binomial coefficient. For example, $\sum_k\binom{n}{k}=(1+1)^n=2^n$. $\sum_k(-1)^k\binom{n}{k}=(1-1)^n=0$. Furthermore, obviously $(x+y)^{n+1}=(x+y)(x+y)^n=x(x+y)^n+y(x+y)^n$. This translates into

$$\sum_k\binom{n+1}{k}x^{n+1-k}y^k=x\sum_k\binom{n}{k}x^{n-k}y^k+y\sum_k\binom{n}{k-1}x^{n+1-k}y^{k-1}.$$

Since the equality of two polynomials in (x,y) means equality of their coefficients, we obtain *Pascal's identity*:

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

This is the defining property for the rows of *Pascal's triangle*, where each term equals the sum of two adjacent terms from the previous row:

$$\begin{array}{ccccccccccccc}
 & & & & 0 & & 1 & & 0 & & & & \\
 & & & 0 & & 1 & & 1 & & 0 & & & \\
 & & 0 & & 1 & & 2 & & 1 & & 0 & & \\
 & 0 & & 1 & & 3 & & 3 & & 1 & & 0 & \\
0 & & 1 & & 4 & & 6 & & 4 & & 1 & & 0\\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1
\end{array}$$

Let us experiment with each row: $1^2 = 1$, $1^2 + 1^2 = 2$, $1^2 + 2^2 + 1^2 = 6$, $1^2 + 3^2 + 3^2 + 1^2 = 20$: we get the numbers marked blue!

`Exercises.` (a) Prove the same for all rows: $\sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n}$. (*Hint:* $(x+y)^{2n} = (x+y)^n(x+y)^n$.)

(b) Prove that $(x+y)^p \equiv x^p + y^p \mod p$ when p is prime. (*Remark:* Note that it follows from Fermat's Little Theorem that $(\widehat{x} + \widehat{y})^p = (\widehat{x} + \widehat{y}) = \widehat{x}^p + \widehat{y}^p$ for all $\widehat{x}, \widehat{y} \in \mathbb{Z}_p$. The claim, however, is that the polynomial expression $(x+y)^n$ in two variables x, y, when its integer coefficients are reduced modulo p, turns into the expression $x^p + y^p$.)

(c) Apply (b) to give a new proof of Fermat's Little Theorem: $a^p \equiv a \mod p$, using induction on $a = 0, 1, 2 \ldots, p-1$.

(d) Define the *exponential power series* as $e^x := 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = \sum_{n \geqslant 0} \frac{x^n}{n!}$, and rearrange the formal power series $e^x e^y$ in two variables into e^{x+y}. (*Hint:* $\frac{(x+y)^n}{n!} = \sum_{k+l=n} \frac{x^k}{k!} \frac{y^l}{l!}$).

4. Counting. In principle, the subject of combinatorics is counting, i.e. answering the question *How many?* about elements of various interesting finite sets. Some such questions are easy to answer.

`Examples` (a) *The product formula.* If the set $S = X \times Y \times \cdots \times Z$ is the Cartesian product of several finite sets, then $|S| = |X| \times |Y| \times \cdots \times |Z|$.

(b) *Power sets.* The set of functions $f : X \to Y$ is denoted Y^X (variant: 2^X when $Y = \{0, 1\}$), because when the sets are finite, the total number of such functions is $|Y|^{|X|}$ (prove it!)

(c) Another counting technique, based on representation of elements of certain sets as "leaves" of a tree with regular branching properties, can be visualized in the context of the following playfully formulated exercise: *An alien has 3 arms with 7 fingers each, and 2 arms with 4 fingers each. How many fingers does the alien have in total?*

Many sets are trickier to count.

`Example:` *Inclusion-exclusion principle.* Let us determine (anew) the number $\varphi(n)$ of integers in $\{1, 2, \ldots, n\}$ relatively prime to n which has, say, exactly three distinct prime factors p, q, and r. Clearly, we have to exclude from the set all multiples of p, q or r. The multiples of p form a fraction $1/p$ of the total size n of the set (and likewise $1/q$ and $1/r$ for the multiples of p and r). Thus, it seems that the answer for $\varphi(n)$ should be

$$n \times \left(1 - \frac{1}{p} - \frac{1}{q} - \frac{1}{r}\right).$$

But some integers are divisible by both, say, p and q, and so we have excluded them twice. Therefore we should include them once (and the same for the multiples of qr and rp). Thus, it seems now that the answer for $\varphi(n)$ should be

$$n \times \left(1 - \frac{1}{p} - \frac{1}{q} - \frac{1}{r} + \frac{1}{pq} + \frac{1}{qr} + \frac{1}{rp}\right).$$

However, some integers are divisible by all the three primes. Each of them was excluded 3 times and then included 3 times. Excluding them once again, we obtain

$$\begin{aligned}\varphi(n) &= n \times \left(1 - \frac{1}{p} - \frac{1}{q} - \frac{1}{r} + \frac{1}{pq} + \frac{1}{qr} + \frac{1}{rp} - \frac{1}{pqr}\right) \\ &= n \times \left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{q}\right)\left(1 - \frac{1}{r}\right)\end{aligned}$$

in agreement with the general number-theoretic formula.

Example: *Stirling's partition numbers* (also known as "Stirling's numbers of the second kind", and denoted $S_{n,r}$) count the number of partitions of the set $\{1, 2, \ldots, n\}$ into r non-empty subsets (in other words, the number of equivalence relations with r equivalence classes on a set of n elements). The number of such partitions, for which $\{n\}$ is one of the subsets, coincides with the number $S_{n-1,r-1}$ of partitions of $\{1, \ldots, n-1\}$ into $r-1$ non-empty subsets. Partitions not containing $\{n\}$ as one of the subsets are obtained by dividing $\{1, \ldots, n-1\}$ into r non-empty subsets, and then adjoining n to one of them. The number of such partitions is $rS_{n-1,r}$. Thus, Stirling's partition numbers satisfy the recursion relation somewhat resembling Pascal's:

$$S_{n,r} = S_{n-1,r-1} + r\, S_{n-1,r}.$$

Yet, no simple closed formula for $S_{n,r}$ is known.

Exercises. (a) Show that $S_{n,1} = S_{n,n} = 1$, for all $n \geqslant 1$, and compute a few rows of "Stirling's triangle".

(b) Prove that $S_{n,2} = 2^n - 1$ for all $n \geqslant 1$.

6. Probability

1. Kolmogorov's axioms. One often encounters experiments (e.g. rolling dice) which, even when repeated under seemingly identical conditions, can lead to different outcomes. Applying the idea of probability to such situations, one assumes that for each outcome there exists a real number p between 0 and 1 such that the fraction of the total number of experiments which lead to this outcome tends to p when the experiment is repeated indefinitely. While applications of probability theory to real life situations rely on this (unverifiable) assumption, the purely mathematical formalism of probability theory can be put on solid logical foundations. This was done in 1933 by Andrey Nikolaevich Kolmogorov, who proposed the following axiomatic approach.

Let S be the *sample space*, i.e. the set of all possible outcomes of a given experiment. By an *event* one means a subset of S which belongs to a certain collection of subsets which form a so-called *σ-algebra*. To explain what it is, let us recall that all subsets of S form a Boolean algebra with respect to the operations of complement, union, and intersection of sets. A σ-algebra must form a *Boolean subalgebra*, i.e. be closed with respect to those operations: together with each subset A, to contain its complement $A^c = S - A$, and together with any finite collection of subsets, to contain their intersection and their union. However, for the purposes of most applications of probability theory one needs to require more: that for any *countable* collection of events, their union and intersection are events too. Thus, by definition, a σ-algebra of events must be some collection of subsets in S which includes $\varnothing, S$, and is closed with respect to passing to complements, as well as taking countable unions and intersections.

Given such a σ-algebra, Kolmogorov's axioms require that there be defined a function, which to each event A (i.e. a subset of S which belongs to the given σ-algebra) associates a real number $P(A)$ (interpreted as probability of the event A) in such a way that the following properties are satisfied:

(1) $0 \leqslant P(A) \leqslant 1$ for all events A
(2) $P(\varnothing) = 0$ and $P(S) = 1$
(3) For any countable collection of *pairwise disjoint*[10] events

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

[10] i.e. $A_i \cap A_j = \varnothing$ for all $i \neq j$.

2. Discrete probability. In the majority of scientific applications, the outcomes (e.g. measurements of a physical quantity) form a continuum (e.g. $\mathbb{R}$, or $\mathbb{R}^n$), making probability theory an applied branch of mathematical analysis. However, when the set $S = \{x_1, x_2, \dots\}$ of outcomes is finite, or even countable, the situation simplifies. Namely, without loss of generality one may assume that the σ-algebra of events consists of all subsets of S. Then the probability function P must associate to each of the one-element events $\{x_i\}$ their probabilities, p_i, which are real numbers between 0 and 1 such that $\sum_{i=1}^{|S|} p_i = 1$, and the probability of an event $A \subset S$ is given by

$$P(A) = \sum_{i:x_i \in A} p_i.$$

`Exercise.` Check that any sequence of non-negative numbers p_i whose sum is 1 determines this way a probability function (also called a *distribution*) satisfying Kolmogorov's axioms.

The assumptions about finiteness or countability of the sample space lead to the theory of *discrete probability*, which in fact, while remaining relatively elementary, captures most phenomena important in applications. When the sample set S is finite, it is often plausible (as, say, in the case of rolling a "fair" die) that the probability distribution is *uniform*, i.e. that all p_i are the same (and hence each equal to $1/|S|$).

Exercises. (a) Suppose that one randomly chooses an integer x and computes its congruence class modulo n. Assuming that the resulting probability distribution on $S = \mathbb{Z}_n$ is uniform, find the probability $P(A)$ of the event that x: is divisible by some divisor d of n; is not divisible by d; is relatively prime to n.

(b) *Monty Hall* hides a prize behind one of three closed doors. You must choose one. Then he opens one of the other two to show you that the prize isn't there. What should you do to maximize your chance of collecting the prize: stick to your first choice, or change your mind?

2. Expected value. A real-valued function $f: S \to \mathbb{R}$ is called a *random variable.* The *expected value* (also called the *mean*, or *mathematical expectation*) of a random variable is defined as the sum of its values weighted by the probabilities:

$$E(f) := \sum_{i:\ x_i \in S} p_i f(x_i).$$

`Example.` The sample set $S = \{1, 2, 3, 4, 5, 6\}$ of a die is a subset of the number line. The identity function (assigning to each outcome its numerical value) is therefore a random variable (let us call it x).

`Exercises.` (a) Assuming that the die is fair, compute: $E(x)$; $E(x^2)$; $E((x - E(x))^2)$.

(b) If (x, y) is the outcome of a fair pair of dice, compute $E(x + y)$.

(c) Let $f, g : S \to \mathbb{R}$ be two random variables. Prove that $E(f + g) = E(f) + E(g)$, and $E(kf) = kE(f)$ for any $k \in \mathbb{R}$. (These properties establish *linearity* of the mean value.)

(d) Give an example showing that generally speaking, the *median value* does not possess such a linearity property.

(e) Does $E(fg)$ always coincide with $E(f)E(g)$?

`Example:` *Gambler's ruin.* A gambler's strategy is successful if the expected dollar amount of his gain is positive. Here is a gambling game: The gambler wins \$1 each time a fair coin comes up heads, and loses \$1 each time it comes up tails. Initially he has $\$n$, and quits when he loses all, or when he reaches his goal of having $\$m$ (where $m \geqslant n$). Let p_n be the probability that starting with $\$n$ the gambler loses (and respectively the probability that he reaches his goal be $1 - p_n$). We want to determine p_n (which should depend on m too).

After the first toss, the gambler has either $\$(n + 1)$ or $\$(n - 1)$ with probability 1/2 each, after which his chances of losing become p_{n+1} and p_{n-1} respectively. Therefore $p_n = p_{n+1}/2 + p_{n-1}/2$, and we get the 2nd order linear recursion relation:

$$p_{n+1} = 2p_n - p_{n-1}.$$

Its characteristic polynomial $\lambda^2 - 2\lambda + 1 = (\lambda - 1)^2$ has a double root $\lambda = 1$. The general solution in this case has the form $p_n = A + Bn$ where A and B are arbitrary constants. To find them, we note that $p_0 = 1$ and $p_m = 0$, because in either case the gambler quits right away, thus having no chance to win in the former case, and to lose in the latter. From these we find $A = 1$ and $B = -1/m$ so that $p_n = 1 - \frac{n}{m}$. When $m > 2n$, the gambler is more likely to lose, and when $m < 2n$ he is more likely to win. The expected value E of the dollar amount at the end of the game gives a quantitative measure of his average success:

$$E := p_n \$0 + (1 - p_n)\$m = \left(1 - \frac{n}{m}\right)\$0 + \frac{n}{m}\$m = \$n.$$

Thus as one might expect, on average the gambler is left with just as much as he invested into the gambling, neither gaining nor losing.

4. Conditional probability. We talk of chances when there is no certainty. But if new information arrives, chances can change. For example, the probability of the event A that the sum $x + y$ on two dice is at least 10, equals 1/6 (check this); but *if* the first die shows 5 (condition C), then the chances for $x + y \geqslant 10$ improve to 1/3. Thus, we have calculated the *conditional probability* $P(A|C) = 1/3$ of event A under the condition C (without even rolling any die). Here is the general definition.

Given some condition $C \subset S$ (i.e. an event which has supposedly occurred), the *conditional probability* $P(A|C)$ of event $A \subset S$ under the condition C is the ratio $P(A \cap C)/P(C)$ of the (unconditional) probabilities $P(A \cap C)$ and $P(C)$.

In our situation of discrete probability, one can describe this by saying that when C occurs, all elementary events outside C become impossible, but the relative weights of all elementary events within C remain unchanged:

$$\forall x \notin C,\ P(\{x\}|C) = 0, \text{ and } \forall x \in C,\ P(\{x\}|C) = \frac{P(\{x\})}{\sum_{y \in C} P(\{y\})}.$$

The denominator in the formula guarantees that $\sum_{x \in S} P(x|C) = 1$, i.e. that the conditional probabilities form a new probability distribution on S, which is actually zero outside C. The relative probability $P(A|C)$ of events in S are simply their probabilities with respect to this new distribution (based on the premise that C occurs).

The above formula is a special case of the so-called *Bayes' theorem*, which says that when the sample space S is partitioned into mutually exclusive events X_i (*cases*) of known positive probabilities $P(X_i)$, and C is some condition whose relative probability $P(C|X_i)$ in each case is known, then the distribution of conditional probabilities between the cases if C occurs, can be found from

$$\forall i, \quad P(X_i|C) = \frac{P(C|X_i)P(X_i)}{\sum_j P(C|X_j)P(X_j)}.$$

Indeed,

$$P(X_i|C) = \frac{P(C \cap X_i)}{P(C)} = \frac{P(C|X_i)P(X_i)}{P(C)},$$

where, since C is the disjoint union of all $C \cap X_j$,

$$P(C) = \sum_j P(C \cap X_j) = \sum_j P(C|X_j)P(X_j).$$

In the simplest situation of just two cases, X and X^c, Bayes' theorem says:

$$P(X|C) = \frac{P(C|X)P(X)}{P(C|X)P(X) + P(C|X^c)P(X^c)},$$

where of course $P(X) + P(X^c) = 1$.

`Example.` Suppose a large group of people is screened for some disease which occurs in, say, 2% of the population. Let T denote the event that the test comes back positive, while D denote the event that the person actually has the disease. The probability $P(D|T)$, that a person who tested positive is actually sick, can be far less than 1 because of *false positive* results, which occur with small probability $P(T|D^c)$ of, say, 3%. Also, there is a chance $P(T^c|D) = 1 - P(T|D)$ of *false negative* outcomes (say, on the scale of 1%), when the screening fails to detect the existing disease. Using Bayes' formula, we find the portion of sick people detected by the test:

$$P(D|T) = \frac{0.99 \cdot 0.02}{0.99 \cdot 0.02 + 0.03 \cdot 0.98} \approx 0.4,$$

and the portion of sick people missed by the test:

$$P(D|T^c) = \frac{0.01 \cdot 0.02}{0.01 \cdot 0.02 + 0.97 \cdot 0.98} \approx 0.0002,$$

where we also applied $P(T^c|D^c) = 1 - P(T|D^c) = 0.97$. It may come as a surprise that in our example only 40% of those who test positive have the disease, so that more thorough and expensive testing will be needed, but only for those who tested positive in the screening, which is about 5% of the population. At the same time, testing negatively improves confidence in health by a hundred fold: from a 2% chance of being sick to 0.02%.

5. Statistical independence. Returning to the Monty Hall problem, let A, B, C be the three doors, and say that you chose A. Assuming that the prize isn't there, you can argue that Monty Hall is forced to open the only door remaining prize-less: If he opens B, the prize must be behind C. The assumption that A is prize-less is true with probability 2/3. Therefore this is your chance to win if you change your choice. If you stick to door A, your chance is $1 - 2/3 = 1/3$.

In fact there is a simpler way to arrive at the same conclusion. It is based on the notion of statistically independent events. Two events A and B are called *independent* if the probability that both occur equals the product of their individual probabilities:

$$P(A \cap B) = P(A)P(B).$$

Assuming that $P(B) > 0$, we find this equivalent to $P(A|B) = P(A)$, i.e. that the chance of A does not change when B occurs.

This is true about your choice of door A and the event that the host opens door B. Indeed, how can any actions of the host of the show *after* the prize is placed change the probability of the prize being behind A? So, when Monty Hall opens door B, it is still true that $P(A|B \text{ opens}) = P(A) = 1/3$, and hence $P(C|B \text{ opens}) = 2/3$.

More generally, several events are *mutually independent* if for any collection of them the probability of their intersection is equal to the product of their probabilities. Say, A, B, C are mutually independent, if they are pairwise independent and $P(A \cap B \cap C) = P(A)P(B)P(C)$.

Exercises. (a) Show that if A and B are independent, then A and B^c are independent.

(b) Let a random integer x belong to every congruence class modulo N with equal probabilities $1/N$, and let m and n be divisors of N. Prove that the events $m|x$ and $n|x$ are independent if and only if m and n are relatively prime. Generalize this to the case of several divisors.

(c) Let $p_1, \dots p_r$ be distinct prime divisors of N. Prove that the events $p_i \nmid x$ are mutually independent, and derive from this the (already familiar) formula for Euler's function:

$$P(x \text{ is relatively prime to } N) := \frac{\varphi(N)}{N} = \prod_{\text{prime } p|N} \left(1 - \frac{1}{p}\right).$$

(d) *Bernoulli trials.* A "loaded" coin turns heads with probability p, and tails with probability $q = 1 - p$ (possibly not equal to 1/2). Assuming that 10 tosses of the coin are mutually independent, find the probability of the outcome $HTTHTTTHTH$.

(e) Show that the probability $P(n, k)$ of the event that k out of n tosses are heads is $\binom{n}{k} p^k q^{n-k}$. (This distribution is called *binomial.*)

(f) Let s be the random variable equal to $+1$ when the toss of a loaded coin turns heads, and -1 when it turns tails. Compute $E(s)$ and $E(s^2)$.

(g) *Independent random variables.* Two (or several) discrete random variables $f, g, \dots, h : S \to \mathbb{R}$ are called (mutually) *independent* if the events $f(x) = a, g(x) = b, \dots, h(x) = c$ are mutually independent for all values of $a, b, \dots, c$. Prove that if f and g are independent, then $E(fg) = E(f)E(g)$. (*Hint:* If a_i are the values of f, and b_j are the values of g, then the events $(f = a_i) \wedge (g = b_j)$ are mutually disjoint.)

(h) *Random walks.* Every second, a drunkard makes one step forward or one backward with equal probabilities and independently of his previous steps. How far from the starting point is he expected to be 100 seconds later? More precisely, find the expected value of the *square* distance $E\left((s_1 + \cdots + s_{100})^2\right)$ (and then take the square root). (*Hint:* By (g), $E(s_i s_j) = 0$ for $i \neq j$.)

7. Graphs and trees

1. Königsberg bridges. Leonard Euler amused himself by trying to walk around town and visit every of its seven bridges exactly once. He always failed, and to find out whether it is possible at all, he transformed the map of Königsberg (Figure 9, left) into the *graph* (Figure 9, right) showing the islands and river banks as *vertices* (A, B, C, D) connected by *edges* (for the bridges a, b, c, d, e, f, g), and — *voila!* — graph theory was born.

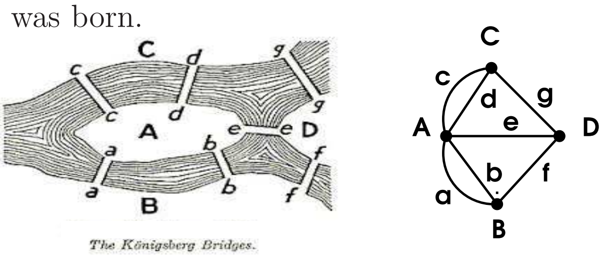

Figure 9

By definition, a *graph* consists of several vertices connected by several edges, and is called *connected* if for any two vertices one can find a chain of edges connecting them.

`Exercises.` (a) Show that if a connected graph has an *Euler circuit* (i.e. a closed chain of edges containing every edge exactly once), then the numbers of edges attached to each vertex[11] are all even.

(b) For each "vertex" of Königsberg, compute the number (it is called the *degree* of the vertex) of the edges attached to it, and derive that this town does not have Euler circuits.

(c) Conversely, prove that if all vertices of a connected graph have even degrees, then Euler circuits exist. (*Hint:* Show that whenever Euler enters a vertex, he can leave it along a yet unwalked edge, until he comes to his starting point. If more than one of such walks are needed to include all edges, argue that the union of these walks can be walked differently as a single circuit.)

(d) A closed circuit in a graph is called *Hamiltonian*, if it visits each vertex exactly once. Show that every Hamiltonian graph with n vertices is obtained from the regular n-gon by adding extra edges.

`Remark.` We should have said "is isomorphic to the graph, obtained ..." Two graphs are called *isomorphic*, if there exist a bijection between the sets of their vertices, and a bijection between the sets of their edges, such that the corresponding edges connect corresponding vertices.

[11] To be accurate, if an edge is a *loop*, meaning that it connects a vertex to itself (which is allowed), then the edge should be considered attached twice to this vertex.

2. The adjacency matrix. A *directed* graph is specified by two finite sets: vertices (V) and edges (E), and by two functions $v_\pm : E \to V$, assigning to each edge $e \in E$ its *head* $v_+(e) \in E$ and its *tail* $v_-(e) \in V$. Pictorially, a directed edge is an arrow pointing from its tail to its head.

The *adjacency matrix* of a directed graph is the square matrix $A = [a_{ij}]$ whose rows and columns are labeled by the vertices of the graph, and whose entry a_{ij} in the ith row and jth column is equal to the number of edges from the vertex j to the vertex i.

Theorem. *The matrix $P(N)$ whose entry $p_{ij}(N)$ is defined as the number of directed walks to vertex i from vertex j consisting of exactly N edges (placed tail-to-head) coincides with the matrix power A^N.*

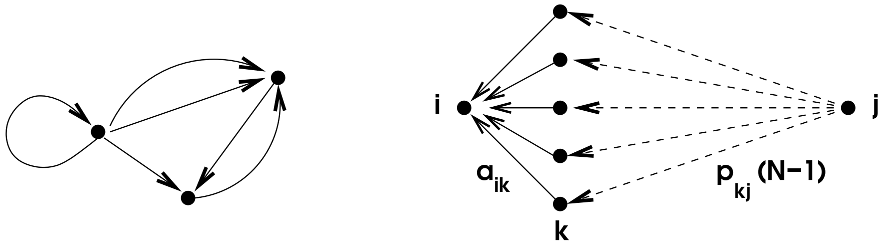

Figure 10

Proof: Induction on N. *Base:* $P(0)$ is the identity matrix A^0. *Step:* For $N > 0$, any N-edge-long walk starting at the vertex j and terminating at the vertex i (Figure 10, right) can come from any of the vertices k, and consist of one of the a_{ik} edges leading from vertex k to vertex i, preceded by any of the $p_{kj}(N-1)$ walks of length $N-1$ leading to vertex k from vertex j. The possibilities of passing through different vertices k are mutually exclusive, hence $p_{ij}(N) = \sum_k a_{ik} \cdot p_{kj}(N-1)$. Miraculously, this is the formula for the elements of the matrix product: $P(N) = A \cdot P(N-1)$. By the induction hypothesis, $P(N-1) = A^{N-1}$, and therefore $P(N) = A^N$. Q.E.D.

Exercises. (a) Write down the adjacency matrix of the graph on Figure 10, left, and compute the numbers of walks of length 4 between its vertices.

(b) Formulate and prove the analogue of the above theorem in the case of undirected graphs (as in Euler's problem). (*Hint:* The matrix A in this case will be symmetric.)

(c) Use (b) and Linear Algebra to compute the total number of closed walks of length N in the *complete graph* with n vertices. (In this graph, every two distinct vertices are connected by an undirected edge.) (*Answer:* $(n-1)^N + (n-1)(-1)^N$.)

3. Spanning trees. An (undirected) connected circuit-free graph is called a *tree*. In other words, in a tree, every two vertices can be connected by a unique *trail* (i.e. a walk with no repeated edges).

Every connected graph has a *spanning tree* (blue on Figure 11, left) i.e. a connected subgraph with the same vertices, which is a tree. Indeed, removing one edge of a circuit leaves the graph connected, and does not change the set of its vertices. Thus, removing such edges one by one as long as circuits remain, we end up with a spanning tree.

`Theorem.` *In a tree, the number of vertices exceeds the number of edges by one:* $|V| - |E| = 1$.

`Proof:` Induction on $|V|$. *Base:* If $|V| = 1$, any edge would be a circuit, so $|E| = 0$. *Step:* A tree with $|V| > 1$ has vertices of degree 1 (they are called *leaves*). Indeed, if all degrees > 1, one can find a circuit (as in Euler's problem). Removing a leaf together with the only adjacent edge does not change $|V|-|E|$, but results in a tree with fewer vertices, for which $|V| - |E| = 1$ by the induction hypothesis. Q.E.D.

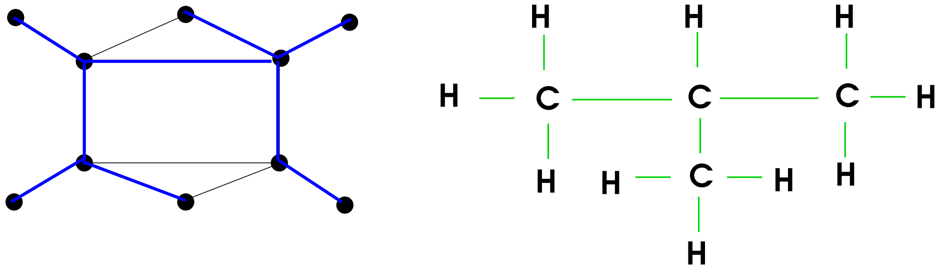

FIGURE 11

`Exercises.` (a) Show that the converse is also true: A connected graph with $|E| = |V| - 1$ is a tree. (*Hint:* Find its spanning tree.)

(b) A *forest* is a disconnected graph, each of whose connected components is a tree. Show that every disconnected graph has a spanning forest, and that $|V| - |E| = \# - |C|$, where $\#$ is the number of connected components and $|C|$ is the number of "independent circuits", i.e. the minimal number of edges that need to be removed to get rid of all circuits.

(c) A hydrocarbon molecule (Figure 11, right) consists of hydrogen and carbon atoms (they can form one, and up to four chemical bonds respectively), and is called *saturated* if the number $|H|$ of hydrogen atoms in it is maximal for a given number $|C|$ of carbon atoms. Prove *Cayley's theorem:* A saturated hydrocarbon molecule must be a tree with $|H| = 2|C| + 2$.

4. Rooted trees. The tree in Figure 12 (left) encodes the syntax of the expression $((a-b)\cdot c)+(d/e)$. It is an example of a full binary rooted tree of height 3, where *rooted* means that one vertex has been designated as *the root*, *height* indicates the maximal distance from the root to the leaves (along a unique trail of edges), *binary* refers to the fact that each *parent* vertex (i.e. not a leaf) has at most two *children* (i.e. adjacent vertices one step farther from the root), and *full* refers to the fact that each parent has exactly two children (designated as *left* and *right*). In general rooted trees don't have to be binary, and binary rooted trees don't have to be "full". They are useful in many situations, e.g. for describing syntax of natural (linguistic) or formal (CS) languages as in this example, but the terminology emphasizes resemblance to genealogical trees.

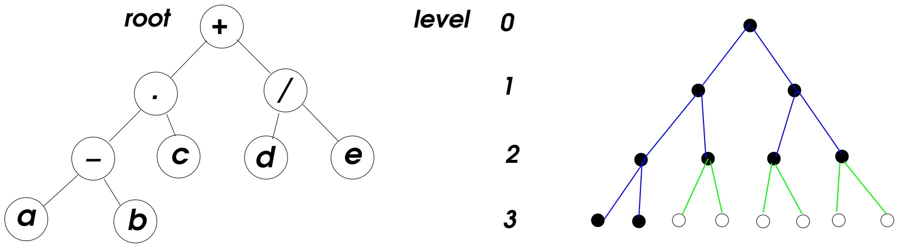

FIGURE 12

`Exercises.` (a) Prove that if a full binary tree has k *internal* vertices (i.e. not leaves), then it has $k+1$ *terminal* vertices (i.e. leaves). (*Hint:* Show that $|E| = 2\times k$.)

(b) Prove that the number t of terminal vertices of any binary tree of height h does not exceed 2^h. (*Hint:* Complete such a tree to *the* full binary tree of height h with exactly 2^h leaves on the bottom level, as in the example of Figure 12, right.)

5. Kruskal's and Prim's algorithms. Suppose the edges of a connected graph are labeled by positive *weights* $w(e)$ (e.g. the cost of transportation of something between cities represented by the vertices), and one needs to find a spanning tree T of minimal total weight $\sum_{e\in T} w(e)$. *Kruskal's algorithm* proposes to build the tree, T_K, by adding edges one at a time in the order of increasing weight, but skipping an edge if it creates a circuit with the previously added ones. *Prim's algorithm* proposes to grow the tree, T_P, from a root vertex, by adding one vertex at a time and always picking the "cheepest" one among the available choices. Figure 13, where the weights $a < b < \cdots < k$ are ordered alphabetically, shows the order in which the edges of the tree are added in Kruskal's (blue), and in Prim's (red and green) algorithms started from the root vertex of that color.

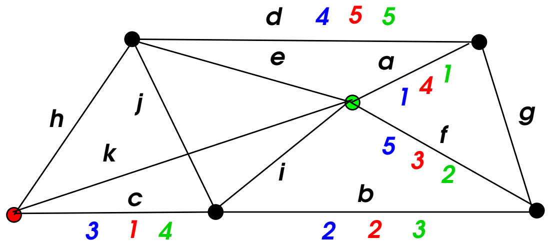

FIGURE 13

`Theorem.` *Kruskal's and Prim's algorithms work correctly.*

`Proof.` (i) To have the total weight less than that of T_K, a tree T must replace edges e_i of T_K of weights $w(e_1) < \cdots < w(e_m) < \dots$ with some edges e'_i of weights $w(e'_1) < \cdots < w(e'_m) < \dots$ such that $w(e'_m) < w(e_m)$ for at least one m. But all edges with $w(e) < w(e_m)$, together with their vertices, form a subgraph where the edges of T_K with $w(e) < w(e_m)$ form a spanning forest. Therefore replacing only $m-1$ edges $e_1, \dots, e_{m-1}$ of this forest with m edges $e'_1, \dots, e'_m$ will create a circuit in T. This contradiction shows that T_K is minimal.

`Remark.` Note that T_K depends not on the weights $w(e)$, but only on the *total ordering* of the edges by their weights. Thus, if some weights are equal, to run the algorithm, one can perturb them a little to remove the coincidence and yet not destroy their ordering relative to other weights. The tree T_K may depend on the perturbation, but the total "unperturbed" weight of T_K will be the same. Now we will prove that T_K coincides with T_P constructed using the same edge ordering.

(ii) Suppose T_P differs from T_K, and let e be an edge in T_P not contained in T_K. On the step of Prim's algorithm when e is added to T_P, it connects a vertex v_- in the previous part of T_P (call it S) with a new vertex v_+ not in S. Adding e to T_K creates a circuit consisting of e and the unique path in T_K from v_- to v_+. Let e' be the first edge of this path which is not in S (it exists, since v_- is in S, but v_+ is not). Then $w(e') > w(e)$, for otherwise e' would have been added to S instead of e on that step of Prim's algorithm. But then, replacing e' in T_K with e, we would get a spanning tree "cheaper" than T_K. This contradicts part (i), and implies that $T_P = T_K$. Q.E.D.

6. Dijkstra's algorithm. Let us interpret the weights $w(e) > 0$ as the edges' lengths, and consider the problem of finding shortest paths from a vertex designated as the *root* of the graph to all other vertices $v \in V$. We may assume (slightly perturbing the weights if needed) that different paths between the same vertices have unequal lengths.

Exercise: *Bellman's principle.* Show that the (unique) shortest path from a given vertex to the root also provides the shortest paths for all intermediate vertices. Deduce that these paths altogether form a spanning (rooted) tree.

Dijkstra's algorithm builds this tree starting from the root. On each step, the algorithms recalculates the function $d : V \to \mathbb{R}^+ \cup \infty$ of *tentative distances* to the root from all vertices $v \in V$ by: (i) selecting from the set $U \subset V$ of yet *unvisited* vertices the one (v_{active}) with the minimal value of d, (ii) for each $v \in U$ adjacent to v_{active}, replacing $d(v)$ with the sum $d(v_{active}) + w(e_{v_{active},v})$ *provided that* this sum is smaller than the current value of $d(v)$, and (iii) removing v_{active} from U. At the initialization, $U := V$, and the value of d is set to 0 for the root vertex, and ∞ for all others. The process stops when $U = \varnothing$.

An example on Figure 14, where weights are marked black, v_{active} red, and visited vertices green, shows the evolution of function d (blue). By storing for each vertex v with $d(v) < \infty$ the last edge $e_{v_{active},v}$ from (ii) which decreased $d(v)$ (shown green), we find Bellman's tree at the end.

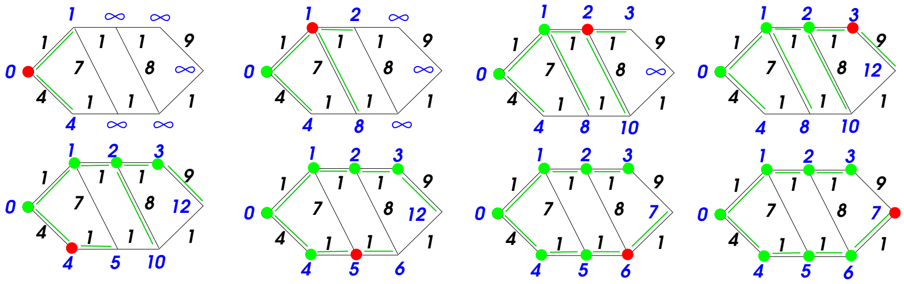

FIGURE 14

Theorem. *Dijkstra's algorithm works correctly.*

Proof. We show by induction that $d(v_{active})$ is the shortest distance from v_{active} to the root. *Base:* When the root is active, this is true. *Step:* On a path from $v_{active} \in U$ to the root, let e_{uv} be an edge connecting $u \in U$ with $v \notin U$. By the induction hypothesis, the minimal distance from (previously active) v to the root equals $d(v)$. Therefore the length of the path $\geqslant d(v) + w(e_{uv}) \geqslant d(u) > d(v_{active})$ since v_{active} provides the minimum of d on U.