

# KG1 notes

Zdeněk Dvořák

January 25, 2026

## Part I

# Introduction to combinatorics

# Lesson 1

## Goals of combinatorial counting and introduction to generating functions

### 1.1 What does it mean to count something?

So far, when we have counted something (permutations, subsets, ...), we have usually seen an exact formula as a result. E.g, the number of subsets of an  $n$ -elements set is  $2^n$ ,  $\binom{n}{k}$  of those have exactly  $k$  elements, and the number of permutations of  $n$  elements without a fixed point (derangements) is

$$n! \cdot \sum_{k=0}^n \frac{(-1)^k}{k!}.$$

However, this may be less desirable than it might first seem. Computing the value of the formula for particular inputs may not be very efficient (what is the time complexity of evaluating the formula for derangements, or even just for computing the binomial coefficient? For how large values of  $n$  is it actually feasible?). Moreover, it is often not easy to determine how large the value expressed by a formula is, or how large it grows. As a warning example, here is an exact formula for the value of the  $n$ -th prime:

$$p_n = 1 + \sum_{i=1}^{2^n} \left[ \left( \frac{n}{\sum_{j=1}^i \left[ \left( \cos \frac{(j-1)!+1}{j} \pi \right)^2 \right]} \right)^{1/n} \right]$$

With a bit of thought, it should be clear that this formula is basically worthless, as evaluating it would take much more time than just computing the  $n$ -th prime by brute force, and using it to show any properties of primes does not seem to be feasible.

So, sometimes some a priori less appealing possibilities which we are going to explore next may actually be more useful. For demonstration, we are going to use the following problem.

**Example 1.** *What is the number  $a_n$  of strings of length  $n$  consisting of letters  $\mathbf{a}$  and  $\mathbf{b}$  such that no two letters  $\mathbf{a}$  are consecutive (let us call such strings  $\mathbf{aa}$ -avoiding)? Thus,  $a_0 = 1$  (the empty string),  $a_1 = 2$  (the strings  $\mathbf{a}$  and  $\mathbf{b}$ ),  $a_2 = 3$  (the strings  $\mathbf{ab}$ ,  $\mathbf{ba}$ , and  $\mathbf{bb}$ ),  $a_3 = 5$  ( $\mathbf{aba}$ ,  $\mathbf{abb}$ ,  $\mathbf{bab}$ ,  $\mathbf{bba}$ , and  $\mathbf{bbb}$ ), and so on.*

### 1.1.1 Recurrence relations

Sometimes, we can find a recurrence relation that enables us to compute the values efficiently.

**Example 2.** *Although there actually exists an exact formula for  $a_n$ , it is not easy to come up with it directly. However, we can see that the  $\mathbf{aa}$ -avoiding strings of length  $n$  which end with  $\mathbf{b}$  are precisely those obtained from the  $\mathbf{aa}$ -avoiding strings of length  $n - 1$  by appending  $\mathbf{b}$ . Moreover, if an  $\mathbf{aa}$ -avoiding string of length  $n \geq 2$  ends with  $\mathbf{a}$ , then it must end with  $\mathbf{ba}$ ; and thus the  $\mathbf{aa}$ -avoiding strings of length  $n \geq 2$  which end with  $\mathbf{a}$  are precisely those obtained from the  $\mathbf{aa}$ -avoiding strings of length  $n - 2$  by appending  $\mathbf{ba}$ . Therefore, we have*

$$a_n = a_{n-1} + a_{n-2}.$$

*Although this does not give an exact formula, we can use this to determine the value of  $a_n$  using only  $n$  additions (so, in a sense, this is about as good as saying “the number of permutations of  $n$  elements is  $n!$ ”, since computing  $n!$  requires  $n$  additions).*

As a remark, you have likely seen this sequence before—indeed, it is the famous *Fibonacci sequence*.

### 1.1.2 Asymptotically precise estimates

When trying to determine the number  $g(n)$  of some objects of size  $n$ , we may be able to obtain a simple expression  $f(n)$  that gives at least an approximate answer. Often a best outcome that we can hope for is getting an *asymptotically precise estimate*, i.e., a simple formula  $f$  such that

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1.$$

**Example 3.** *The famous Stirling’s formula has this property:*

$$\lim_{n \rightarrow \infty} \frac{n!}{\sqrt{2\pi n}(n/e)^n} = 1.$$

Thus, for every  $\varepsilon > 0$ , there exists  $n_0$  such that

$$(1 - \varepsilon) \cdot \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \leq n! \leq (1 + \varepsilon) \cdot \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

for every  $n \geq n_0$ . E.g., for sufficiently large  $n$ , the expression  $\sqrt{2\pi n}(n/e)^n$  differs from  $n!$  by less than 1% (in this particular example,  $n_0 = 9$  suffices). Moreover, this expression can be computed using  $O(\log n)$  arithmetic operations, while computing  $n!$  directly takes  $n$  multiplications.

### 1.1.3 Growth rates

Even less precise approximations are often useful.

**Example 4.** As in Example 1, let  $a_n$  be the number of  $\mathbf{aa}$ -avoiding strings of length  $n$ , and recall that  $a_n = a_{n-1} + a_{n-2}$  for every  $n \geq 2$ ,  $a_0 = 1$ , and  $a_1 = 2$ . Let  $\gamma$  be the positive real number such that  $\gamma^2 = \gamma + 1$  ( $\gamma = (1 + \sqrt{5})/2 \approx 1.618$ ). Then

$$\gamma^n \leq a_n \leq 2\gamma^n.$$

*Solution.* Note that  $\gamma^0 = a_0 < 2\gamma^0$  and  $\gamma^1 < a_1 < 2\gamma^1$ . We now proceed by induction;  $n \geq 2$ , we have

$$\begin{aligned} a_n &= a_{n-1} + a_{n-2} \\ &\leq 2\gamma^{n-1} + 2\gamma^{n-2} = 2\gamma^{n-2} \cdot (1 + \gamma) \\ &= 2\gamma^{n-2} \cdot \gamma^2 = 2\gamma^n. \end{aligned}$$

The lower bound is proved analogously. ■

In other words, we have  $a_n = \Theta(\gamma^n)$ . This estimate is not asymptotically precise, since the upper and lower bounds differ by a constant factor; nevertheless, it gives us a very good idea of how fast the sequence  $a_n$  grows.

### 1.1.4 Bijections

Finally, it may be useful to directly relate two quantities, even if we cannot (yet) compute either of them.

**Example 5.** For  $n \geq 2$ , let  $b_n$  be the number of ways how to express  $n$  as a sum of integers greater than one, where the order of the terms in the sum matters. Thus, we have  $b_2 = 1$  (we can only use the trivial sum consisting of just the term 2),  $b_3 = 1$  (the only possibility is the trivial single-term sum 3),  $b_4 = 2$  (there are two possibilities, 2 + 2 or 4),  $b_5 = 3$  (the possibilities are 2 + 3, 3 + 2, and 5), and  $b_6 = 5$  (the possibilities are 2 + 2 + 2, 2 + 4, 3 + 3, 4 + 2, and 5). For  $n \geq 3$ , we have  $b_n = a_{n-3}$ .

*Solution.* Imagine we have such a sum  $n = s_1 + \dots + s_k$ . We can represent it as follows: We write down  $s_1 - 1$  letters  $\mathbf{b}$  and one letter  $\mathbf{a}$ , then  $s_2 - 1$  letters

**b** and one letter **a**, and so on, obtaining a string of letters **a** and **b** of length  $n$ . Moreover, since  $s_1, \dots, s_k \geq 2$ , this string is clearly **aa**-avoiding!

This mapping from the sums to **aa**-avoiding strings is injective (given the resulting string, we can uniquely decode the original sum), but it is not quite a bijection: Every string obtained from a sum starts with **b** and ends in **ba**. However, observe that this is the only constraint; and moreover, **aa**-avoiding strings of length  $n \geq 3$  starting with **b** and ending in **ba** are exactly obtained from all **aa**-avoiding strings of length  $n - 3$  by prepending **b** and appending **ba**. Thus, for  $n \geq 3$ , the number of ways how to express  $n$  as a sum of integers greater than one is exactly equal to the number of **aa**-avoiding strings of length  $n - 3$ , i.e.,  $b_n = a_{n-3}$ . ■

In particular, if we somehow manage to get a formula for  $a_n$ , we also get one for  $b_n$  for free, and vice versa.

## 1.2 Generating functions

Somewhat surprisingly, some of the most powerful tools for combinatorial counting (a discrete problem) use continuous mathematics. One reason for that is the notion of generating functions, which turns out to be immensely useful in achieving all the goals outlined in the previous section. The *generating function* of a sequence  $s_0, s_1, s_2, \dots$  (typically of non-negative integers) is the function  $S : \mathbb{R} \rightarrow \mathbb{R}$  defined by

$$S(x) = \sum_{n=0}^{\infty} s_n x^n$$

for all real numbers  $x$  such that this series converges.

**Example 6.** Consider the sequence  $1, 2, 4, 8, \dots$ , i.e., the sequence with  $n$ -th term  $p_n = 2^n$ . The generating function of this sequence is

$$1 + 2x + 4x^2 + 8x^3 + \dots = (2x)^0 + (2x)^1 + (2x)^2 + (2x)^3 + \dots = \frac{1}{1 - 2x},$$

where the last equality is the well-known formula for the sum of the geometric series (and is valid whenever  $|2x| < 1$ , i.e., when  $|x| < 1/2$ ).

More generally, for every real number  $r$ , the generating function of the sequence  $r^0, r^1, r^2, \dots$  is  $\frac{1}{1-rx}$ .

Generating functions in combinatorial counting are typically applied when we are interested in how many objects of some kind (strings, trees, ...) are there of given size  $n$ . To illustrate, as in Example 1, let  $a_n$  be the number of **aa**-avoiding strings of length  $n$ . The generating function of the sequence  $a_0, a_1, a_2, \dots$  is

$$A(x) = \sum_{n=0}^{\infty} a_n x^n = 1 + 2x + 3x^2 + 5x^3 + \dots \quad (1.1)$$

Let us for the moment ignore the question of convergence of this series and proceed a bit haphazardly (we will make sure we are formally on a safe ground in the following lecture). At first glance, defining this generating function for  $a_n$  may seem a bit pointless. Given that we do not understand this sequence all that well, it is difficult to imagine we would be able to evaluate  $A(x)$ —and in fact, doing so would be useless, since the values of the generating function do not have much meaning. However, it turns out that using the recurrence  $a_n = a_{n-1} + a_{n-2}$ , we can determine the function  $A$  precisely. Indeed, consider the expression  $(1 - x - x^2)A(x) = A(x) - xA(x) - x^2A(x)$ . By Equation 1.1, we can write this expression as follows (by the recurrence, we have  $a_n - a_{n-1} - a_{n-2} = 0$  for every  $n \geq 2$ ):

$$\begin{array}{rcccccc}
 & a_0 & & +a_1x & & +a_2x^2 & & +a_3x^3 & +\dots \\
 & & & -a_0x & & -a_1x^2 & & -a_2x^3 & -\dots \\
 & & & & & -a_0x^2 & & -a_1x^3 & -\dots \\
 = & a_0 & + & (a_1 - a_0)x & + & (a_2 - a_1 - a_0)x^2 & + & (a_3 - a_2 - a_1)x^3 & +\dots \\
 = & a_0 & + & (a_1 - a_0)x & & & & & \\
 = & 1 & & +x & & & & & 
 \end{array}$$

Thus,  $(1 - x - x^2)A(x) = (1 + x)$ , and consequently

$$A(x) = \frac{1 + x}{1 - x - x^2}.$$

When we manage to obtain an explicit formula for the generating function, mathematical analysis gives us many tools to obtain good estimates on the coefficients  $a_n$  of its Taylor series. In some cases, we can even obtain an exact formula!

In particular, this is always the case when the generating function is *rational*, i.e., a ratio of two polynomials. The basic trick here is that any rational function  $p(x)/q(x)$  with  $\deg p < \deg q$  can be expressed as a linear combination of simple functions of form  $\frac{1}{(x-t)^k}$ , where  $t$  is a root of the polynomial  $q$  and  $k$  is a positive integer at most as large as the multiplicity of the root  $t$  (you should be familiar with this idea from mathematical analysis, where it is used to compute integrals of rational functions). In particular, in the simplest case that all roots of  $q$  have multiplicity one, all terms of this linear combination will be of form  $\frac{1}{x-t}$ . Note that this is very similar to the functions considered in Example 6; and to make this similarity even more clear, note that we can divide by  $-t$  the denominator of this fraction (as well as the coefficient with which this fraction appears in the linear combination) to change it to  $\frac{1}{1-t^{-1}x}$ . Skipping the technical computations, we have

$$\begin{aligned}
\sum_{n=0}^{\infty} a_n x^n &= A(x) = \frac{1+x}{1-x-x^2} \\
&= \frac{5+3\sqrt{5}}{10} \cdot \frac{1}{1-\frac{\sqrt{5}+1}{2}x} + \frac{5-3\sqrt{5}}{10} \cdot \frac{1}{1-\frac{1-\sqrt{5}}{2}x} \\
&= \frac{5+3\sqrt{5}}{10} \cdot \sum_{n=0}^{\infty} \left(\frac{\sqrt{5}+1}{2}x\right)^n \\
&\quad + \frac{5-3\sqrt{5}}{10} \cdot \sum_{n=0}^{\infty} \left(\frac{1-\sqrt{5}}{2}x\right)^n \\
&= \sum_{n=0}^{\infty} \left( \frac{5+3\sqrt{5}}{10} \cdot \left(\frac{\sqrt{5}+1}{2}\right)^n + \frac{5-3\sqrt{5}}{10} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n \right) x^n.
\end{aligned}$$

Hence, by comparing the coefficients at  $x^n$ , we obtain the following curious formula:

$$a_n = \frac{5+3\sqrt{5}}{10} \cdot \left(\frac{\sqrt{5}+1}{2}\right)^n + \frac{5-3\sqrt{5}}{10} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Note that  $\left|\frac{1-\sqrt{5}}{2}\right| < 1$ , and thus the second term in this sum becomes negligibly small as  $n$  goes to infinity. Thus,

$$a_n \approx \frac{5+3\sqrt{5}}{10} \cdot \left(\frac{\sqrt{5}+1}{2}\right)^n,$$

where the difference between the left and the right hand side limits to 0 as  $n$  goes to infinity. It is of course not a coincidence that  $\frac{\sqrt{5}+1}{2}$  is precisely the constant  $\gamma$  which appeared in Example 4.

### 1.3 Homework

1. For a non-negative integer  $n$ , let  $c_n$  be the number of strings of length  $n$  consisting of the letters **a**, **b**, and **c** that do not contain the substrings **aa**, **ab**, **ba**, and **bb**. Thus,  $c_0 = 1$ ,  $c_1 = 3$ ,  $c_2 = 5$ , and  $c_3 = 11$ . Show that

$$c_n = c_{n-1} + 2c_{n-2}$$

holds for every  $n \geq 2$ .

2. Show that  $c_n = \Theta(2^n)$ .
3. Show that the following two quantities are equal for every  $n \geq 1$ :
  - The number of strings of length  $n$  consisting of letters **a** and **b** and starting with **a**.
  - The number of ways to express  $n$  as a sum of positive integers, where the order of the terms in the sum matters.

## 1.4 Tutorial

1. Find explicit formulas for generating functions of the following sequences:
  - 1, 2, 5, 0, 0, 0, ...
  - 1, 1, 1, 1, ...
  - 0, 1, 0, 1, ...
  - 2, -3, 2, -3, ...
  - 0, 1, 2, 3, 4, ... (hint: use derivatives)
  - $0 \cdot 2^0, 1 \cdot 2^1, 2 \cdot 2^2, 3 \cdot 2^3, 4 \cdot 2^4, \dots$
  - $0^2, 1^2, 2^2, 3^2, 4^2, \dots$
  - $2^2, 3^2, 4^2, 5^2, 6^2, \dots$
2. Show that if  $B(x)$  is the generating function of the sequence  $b_0, b_1, b_2, \dots$ , then  $\frac{B(x)}{1-x}$  is the generating function of the sequence  $b_0, b_0 + b_1, b_0 + b_1 + b_2, \dots$
3. Use this to derive a formula for the sum  $\sum_{k=0}^n 3^k$ .
4. Suppose  $e_0, e_1, e_2, \dots$  is a sequence such that  $e_0 = 5, e_1 = 0$ , and  $e_n = e_{n-1} + 6e_{n-2}$  for every  $n \geq 2$ . Find an explicit expression for the generating function of this sequence and use it to find an explicit formula for  $e_n$ .
5. Denote by  $t_n$  the number of ways to cover a  $2 \times n$  rectangle with  $2 \times 1, 1 \times 2$ , and  $2 \times 2$  domino pieces. Determine a linear recurrence and generating function for  $t_n$  and use it to find an explicit formula.
6. Show that tilings of the  $2 \times n$  rectangle are in bijection with tilings of a  $1 \times n$  board by one type of length-1 tile and two distinct types of length-2 tiles.

## Lesson 2

# Theory underlying generating functions and operations on generating functions

### 2.1 Correctness of generating function usage

In the last lecture, we have seen a simple application of generating functions in combinatorics, to compute the number of **aa**-avoiding strings of given length  $n$  (i.e., strings consisting of letters **a** and **b** that do not contain two consecutive **a**'s). If you think through the example, you might be somewhat uneasy about the correctness of all the steps involving infinite series. Let us go through the example again, building up the theory a bit more carefully.

Recall that a generating function of a sequence  $a_0, a_1, \dots$  is the function  $A(x) = \sum_{n=0}^{\infty} a_n x^n$ . We usually apply this concept to compute the number of objects of certain kind and size. More formally, a *combinatorial class* is a set  $\mathcal{C}$  together with a function  $\text{size}_{\mathcal{C}} : \mathcal{C} \rightarrow \mathbb{N}$  such that  $\text{size}_{\mathcal{C}}^{-1}(n)$  is finite for each non-negative integer  $n$ . The *counting sequence* of  $\mathcal{C}$  is the sequence  $c_0, c_1, \dots$  such that for each  $n$ , the term  $c_n = |\text{size}_{\mathcal{C}}^{-1}(n)|$  is the number of objects in  $\mathcal{C}$  of size  $n$ . The *generating function* of a combinatorial class  $\mathcal{C}$  is defined as the generating function of its counting sequence.

**Example 7.** Let  $\mathcal{A}$  be the set of all **aa**-avoiding strings, and for  $s \in \mathcal{A}$ , let us define  $\text{size}_{\mathcal{A}}(s)$  to be the length of the string  $s$ . This turns  $\mathcal{A}$  into a combinatorial class. The counting sequence of  $\mathcal{A}$  is the sequence  $a_0, a_1, \dots$  where  $a_n$  is the number of **aa**-avoiding strings of length  $n$ , and the generating function of  $\mathcal{A}$  is

$$A(x) = \sum_{n=0}^{\infty} a_n x^n = 1 + 2x + 3x^2 + 5x^3 + \dots$$

for all  $x$  such that this series converges.

We have observed that the sequence satisfies the recurrence relation  $a_n = a_{n-1} + a_{n-2}$  for every  $n \geq 2$ , and used it to argue about the generating function  $A(x)$ . First, we proved that the equality  $A(x) = \frac{1+x}{1-x-x^2}$  must hold for all  $x$  such that  $A(x)$  converges:

$$\begin{array}{rcccc}
 & & & & +a_3x^3 & +\cdots \\
 & & & & -a_2x^3 & -\cdots \\
 & & & & -a_1x^3 & -\cdots \\
 & & & & -a_0x^2 & \\
 & & & & -a_1x^2 & \\
 & & & & -a_2x^2 & \\
 & & & & +a_3x^2 & \\
 & & & & +a_2x^2 & \\
 & & & & +a_1x & \\
 & & & & +a_0x & \\
 = & a_0 & + & (a_1 - a_0)x & + & (a_2 - a_1 - a_0)x^2 & + & (a_3 - a_2 - a_1)x^3 & + & \cdots \\
 = & a_0 & + & (a_1 - a_0)x & & & & & & \\
 = & 1 & & + & x & & & & & 
 \end{array}$$

We then calculated that

$$\frac{1+x}{1-x-x^2} = \frac{5+3\sqrt{5}}{10} \cdot \frac{1}{1-\frac{\sqrt{5}+1}{2}x} + \frac{5-3\sqrt{5}}{10} \cdot \frac{1}{1-\frac{1-\sqrt{5}}{2}x}$$

for all  $x \neq \frac{-1 \pm \sqrt{5}}{2}$ . Finally, we expanded the right-hand side using the summation formula for geometric series to show that

$$\frac{1+x}{1-x-x^2} = \sum_{n=0}^{\infty} a'_n x^n$$

for all  $x$  such that this series converges, where

$$a'_n = \frac{5+3\sqrt{5}}{10} \cdot \left(\frac{\sqrt{5}+1}{2}\right)^n + \frac{5-3\sqrt{5}}{10} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

From this, we concluded that  $a_n = a'_n$  must hold for every non-negative integer  $n$ .

There are several issues we might feel uneasy about:

- Is the series defining the generating function  $A(x)$  actually convergent for any  $x \neq 0$ ?
- Are the manipulations we performed with the infinite series valid?
- Does the fact that  $\sum_{n=0}^{\infty} a_n x^n = \sum_{n=0}^{\infty} a'_n x^n$  for some range of values of  $x$  actually imply that  $a_n = a'_n$  must hold for all  $n$ ?

Note that we can often sidestep these issues by treating the generating functions as “formal series”, i.e., just as fancy ways how to write down the sequence and not as actually defining a function. However, we cannot avoid them completely if we want to use tools from mathematical analysis. The following standard result puts us on a safe ground.

**Theorem (Safety Theorem).** Let  $s_0, s_1, s_2, \dots$  be a sequence of real numbers such that  $r = \limsup_{n \rightarrow \infty} \sqrt[n]{|s_n|}$  is finite, and let  $R = \frac{1}{r}$  (where  $\frac{1}{0} = \infty$ ). Then the series  $S(x) = \sum_{n=0}^{\infty} s_n x^n$  is convergent for all complex numbers  $x$  such that  $|x| < R$ .

We say that  $R \in \mathbb{R}^+ \cup \{\infty\}$  is the radius of convergence of the series  $\sum_{n=0}^{\infty} s_n x^n$ .

Moreover:

- Let  $t_0, t_1, t_2, \dots$  be another sequence of real numbers such that the series  $T(x) = \sum_{n=0}^{\infty} t_n x^n$  has radius of convergence  $R' > 0$ . If  $(t_0, t_1, \dots) \neq (s_0, s_1, \dots)$ , then there exists  $x$  such that  $0 < x < \min(R, R')$  and  $S(x) \neq T(x)$ .
- Suppose  $f : \mathbb{C} \rightarrow \mathbb{C}$  is a function equal to  $S(x)$  for all  $x \in \mathbb{C}$  such that  $|x| < R$ . Then the complex derivative  $f'(x)$  is defined for all  $x \in \mathbb{C}$  such that  $|x| < R$ . Moreover, if  $s_n \geq 0$  for every  $n$ , then for every  $\varepsilon > 0$ , there exists  $x \in \mathbb{C}$  such that  $|x - R| < \varepsilon$  and either  $f(x)$  or  $f'(x)$  is not defined.

**Example 8.** Note that  $a_n$  is the number of some of strings of length  $n$  composed of letters **a** and **b**, and thus  $a_n \leq 2^n$ . It follows that  $\limsup_{n \rightarrow \infty} \sqrt[n]{a_n} \leq 2$ , and thus the series defining the generating function  $A(x) = \sum_{n=0}^{\infty} a_n x^n$  has radius of convergence  $R \geq \frac{1}{2}$  and converges for all  $x \in \mathbb{C}$  such that  $|x| < R$ .

The calculation showing that  $A(x) = \frac{1+x}{1-x-x^2}$  is correct for all  $x$  such that the series converges (i.e., for all  $x \in \mathbb{C}$  such that  $|x| < R$ ), since it only uses the fact that we can add convergent series term-by-term. At this point, we know enough to determine the radius of convergence (we don't need to, but it gives us useful information about the growth rate of the series).

**Example 9.** The radius of convergence of  $A(x)$  is  $R = \frac{\sqrt{5}-1}{2}$ .

*Solution.* Let  $f(x) = \frac{1+x}{1-x-x^2}$  for all  $x \in \mathbb{C}$ . By the standard rule for derivative of a fraction, we have  $f'(x) = \frac{p(x)}{(1-x-x^2)^2}$  for a polynomial  $p$ . When  $1-x-x^2 \neq 0$ , both  $f(x)$  and  $f'(x)$  are defined. Thus, the only points where they are not defined are the roots  $\frac{-1 \pm \sqrt{5}}{2}$  of  $1-x-x^2$ . Therefore, the last part of the Safety Theorem implies that  $R$  is equal to the smallest positive root  $\frac{\sqrt{5}-1}{2}$  of this polynomial. ■

This already gives us useful information about the growth rate of  $a_n$ . Indeed, we have the following useful observation, which is just the restatement of the fact that  $R = \limsup_{n \rightarrow \infty} \sqrt[n]{|a_n|}$ .

**Observation 10.** Let  $S(x) = \sum_{n=0}^{\infty} s_n x^n$  and let  $R$  be the radius of convergence of this series.

- For every  $\varepsilon > 0$ , there exists  $n_0$  such that  $|s_n| \leq \left(\frac{1}{R} + \varepsilon\right)^n$  for every  $n \geq n_0$ .

- For every  $\varepsilon > 0$ , there exist infinitely many values of  $n$  such that  $|s_n| \geq \left(\frac{1}{R} - \varepsilon\right)^n$ .

In our case, we have  $\frac{1}{R} = \frac{2}{\sqrt{5}-1} = \frac{\sqrt{5}+1}{2}$ , and thus Observation 10 tells us that for every  $\varepsilon > 0$ , we have  $a_n \leq \left(\frac{\sqrt{5}+1}{2} + \varepsilon\right)^n$  for all sufficiently large  $n$  (of course, we already know that from Example 4).

Finally, the claim that  $a_n = a'_n$  follows from the fact that

$$\sum_{n=0}^{\infty} a_n x^n = \frac{1+x}{1-x-x^2} = \sum_{n=0}^{\infty} a'_n x^n$$

holds for all  $x$  such that both series converge (and both of the series have non-zero radius of convergence).

In the rest of the presentation, we are going back to the less formal style, not bothering to state all the technical details. With the ideas you just learned, you can easily check that we are not cheating there, either.

## 2.2 Operations on generating functions

In the example we have shown, we found the explicit expression for the generating function somewhat ad-hoc. Importantly, there turns out to be a quite systematic way to go about this, based on the fact that combinatorial operations correspond to algebraic operations on the generating functions. To illustrate, let us start with the following nearly obvious claim.

**Observation 11.** *Let  $\mathcal{A}$  and  $\mathcal{B}$  be disjoint combinatorial classes and let  $A(x)$  and  $B(x)$  be their generating functions. Then their union  $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$  has generating function  $C(x) = A(x) + B(x)$ .*

*Proof.* Let  $a_0, a_1, \dots$  and  $b_0, b_1, \dots$  be the counting sequences of  $\mathcal{A}$  and  $\mathcal{B}$ , respectively. For each  $n$ ,  $\mathcal{C}$  contains  $a_n + b_n$  objects of size  $n$ , and thus its counting sequence  $c_0, c_1, \dots$  satisfies  $c_n = a_n + b_n$ . Hence,

$$C(x) = \sum_{n=0}^{\infty} c_n x^n = \sum_{n=0}^{\infty} (a_n + b_n) x^n = \sum_{n=0}^{\infty} a_n x^n + \sum_{n=0}^{\infty} b_n x^n = A(x) + B(x).$$

□

More importantly, multiplication of generating functions also has combinatorial meaning. For combinatorial classes  $\mathcal{A}$  and  $\mathcal{B}$ , their *cartesian product*  $\mathcal{A} \times \mathcal{B}$  is the combinatorial class  $\{(a, b) : a \in \mathcal{A}, b \in \mathcal{B}\}$  such that  $\text{size}_{\mathcal{A} \times \mathcal{B}}(a, b) = \text{size}_{\mathcal{A}}(a) + \text{size}_{\mathcal{B}}(b)$ .

**Lemma 12.** *Let  $\mathcal{A}$  and  $\mathcal{B}$  be combinatorial classes and let  $A(x)$  and  $B(x)$  be their generating functions. Then  $\mathcal{D} = \mathcal{A} \times \mathcal{B}$  has generating function  $D(x) = A(x)B(x)$ .*

*Proof.* Let  $a_0, a_1, \dots, b_0, b_1, \dots$ , and  $d_0, d_1, \dots$  be the counting sequences of  $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\mathcal{D}$ , respectively. In how many ways can we choose elements  $a \in \mathcal{A}$  and  $b \in \mathcal{B}$  such that  $\text{size}_{\mathcal{A}}(a) + \text{size}_{\mathcal{B}}(b) = n$ ? We can pick  $a$  of size 0 and  $b$  of size  $n$  (there are  $a_0 b_n$  possible ways how to do it), or  $a$  of size 1 and  $b$  of size  $n - 1$  (there are  $a_1 b_{n-1}$  ways), or  $\dots$  Thus,

$$d_n = \sum_{k=0}^n a_k b_{n-k},$$

and

$$\begin{aligned} D(x) &= \sum_{n=0}^{\infty} d_n x^n = \sum_{n=0}^{\infty} \left( \sum_{k=0}^n a_k b_{n-k} \right) x^n \\ &= \sum_{n=0}^{\infty} \sum_{k=0}^n (a_k x^k) \cdot (b_{n-k} x^{n-k}) \\ &= \left( a_0 x^0 \sum_{m=0}^{\infty} b_m x^m \right) + \left( a_1 x^1 \sum_{m=0}^{\infty} b_m x^m \right) + \left( a_2 x^2 \sum_{m=0}^{\infty} b_m x^m \right) + \dots \\ &= \sum_{k=0}^{\infty} a_k x^k \cdot \sum_{m=0}^{\infty} b_m x^m = A(x) \cdot B(x); \end{aligned}$$

Here, we need to know that we can multiply two convergent infinite sums with non-negative terms by summing all pairs of products of their terms, a standard result from mathematical analysis.  $\square$

Of course, we can also obtain a generating function for the number of triples by taking the product of the three generating functions for its elements, etc. This is all a bit abstract, so let us work out another important example.

**Example 13.** Let  $c_n$  be the number of strings of opening and closing brackets of length  $2n$  that are correctly matched, i.e., there are exactly  $n$  opening brackets and  $n$  closing ones, and each prefix of the string contains at least as many opening brackets as the closing ones. E.g.,  $c_0 = 1$  (the empty string “”),  $c_1 = 1$  (the string “()”),  $c_2 = 2$  (the strings “(())” and “()()”), and  $c_3 = 5$  (the strings “((()))”, “(()())”, “(())()”, “()()()”, and “()()()”).

Thus, we are dealing with the combinatorial class  $\mathcal{C}$  consisting of all correctly matched strings of brackets, with  $\text{size}_{\mathcal{C}}(s)$  being equal to half the length of  $s$ ; let  $C$  be its generating function. This combinatorial class contains one “degenerate” element, namely the empty string. Any non-empty string from  $\mathcal{C}$  starts with an opening bracket, which has to be matched with some closing bracket later in the string. Thus, we have  $s = (s_1)s_2$ , where  $s_1$  and  $s_2$  are (possibly empty) correctly matched strings of brackets. In other words, there is a bijection between non-empty strings  $s \in \mathcal{C}$  and the triples (“()”,  $s_1, s_2$ ), where  $s_1, s_2 \in \mathcal{C}$  and

$$\text{size}_{\mathcal{C}} s = 1 + \text{size}_{\mathcal{C}}(s_1) + \text{size}_{\mathcal{C}}(s_2) = \text{size}_{\mathcal{C}}(“()”) + \text{size}_{\mathcal{C}}(s_1) + \text{size}_{\mathcal{C}}(s_2).$$

Equivalently, there is a bijection between  $\mathcal{C}$  and  $\{\text{“”}\} \cup \{\text{“()”}\} \times \mathcal{C} \times \mathcal{C}$ . The combinatorial class  $\{\text{“”}\}$  has generating function 1 and the combinatorial class  $\{\text{“()”}\}$  has generating function  $x$ . Thus, by Observation 11 and Lemma 12, we obtain the following equation for the generating function of  $\mathcal{C}$ :

$$C(x) = 1 + xC^2(x).$$

This gives a quadratic equation for  $C$ , which has two solutions

$$C(x) = \frac{1 \pm \sqrt{1 - 4x}}{2x}.$$

The solution with plus sign does not work, since we also know that the generating function  $C$  must satisfy  $C(0) = 0$  (and we cannot “switch” between the two solutions later, since it is also easy to check that a generating function is continuous between 0 and its radius of convergence). Therefore, we have

$$C(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

This looks like a tougher generating function to handle than the rational function from the **aa**-avoiding strings example. Indeed, it is not quite clear what to do with the expression  $\sqrt{1 - 4x} = (1 - 4x)^{1/2}$ . However, there is a very useful result that enables us to handle it, the Generalized binomial formula; we are going to discuss it in the following lecture.

## 2.3 Homework

1. Let  $\mathcal{B}$  be a combinatorial class and suppose that  $\mathcal{B}$  has generating function

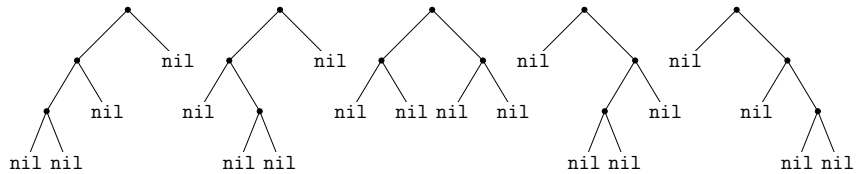
$$B(x) = \frac{1 + x^5}{1 - 3x - x^2}.$$

Determine the radius of convergence of this generating function. What bounds does this give for the counting sequence of  $\mathcal{B}$ ?

2. Let  $\mathcal{S}$  be the combinatorial class of all sums whose terms are integers greater than one (order of terms matters); thus,  $\mathcal{S}$  contains for example the empty sum "", the single-term sum "2", and the sums "2 + 2 + 3", "2 + 3 + 2", and "2 + 2 + 3". We define the size of a sum to be its value; e.g.,  $\text{size}_{\mathcal{S}}("2 + 3 + 2") = 7$ . Let  $\mathcal{S}_1 = \{"2", "3", \dots\}$  be the combinatorial class of all single-term sums from  $\mathcal{S}$ .

- (a) Show that the generating function of  $\mathcal{S}_1$  is  $\frac{x^2}{1-x}$ .  
 (b) Show that  $\mathcal{S}$  is isomorphic to  $\{"\}\cup \mathcal{S}_1 \times \mathcal{S}$ .  
 (c) Translate (b) to an algebraic relation for the generating function  $S(x)$  of  $\mathcal{S}$ .

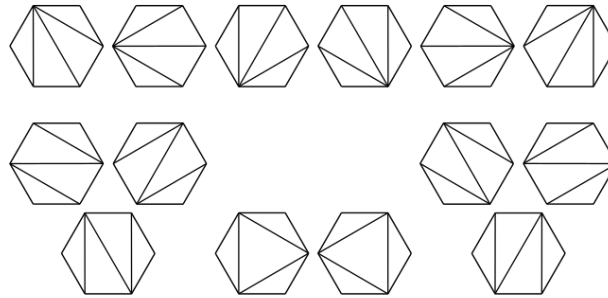
3. Binary trees are recursively defined as follows: A *binary tree* either is the empty tree `nil`, or consists of the root vertex, the left son, and the right son, where both sons are also binary trees. The number of vertices of `nil` is 0, and the number of vertices of a non-empty binary tree with left son  $l$  and right son  $r$  is  $1 + n_l + n_r$ , where  $n_l$  is the number of vertices of  $l$  and  $n_r$  is the number of vertices of  $r$ . Thus, these are the binary trees with three vertices:



Show that for every non-negative integers  $n$ , the number of binary trees with  $n$  vertices is equal to the number of correctly matched strings of opening and closing brackets of length  $2n$ .

## 2.4 Tutorial

1. For  $n \geq 1$ , let  $p_n$  be the number of non-decreasing sequences of integers  $x_1, \dots, x_n$  such that  $x_1 = 1, x_n = n$ , and  $x_i \leq i$  for every  $i \in \{2, \dots, n-1\}$ . For instance,  $p_4 = 5$  counts the following sequences:  $1, 1, 1, 4$ ;  $1, 1, 2, 4$ ;  $1, 1, 3, 4$ ;  $1, 2, 2, 4$ ; and  $1, 2, 3, 4$ . Show that  $t_n$  is equal to the number of correctly matched strings of opening and closing brackets of length  $2n - 2$  by finding a bijection.
2. For  $n \geq 3$ , let  $q_n$  be the number of ways how we can cut a convex  $n$ -gon into triangles by  $n - 3$  non-crossing line segments. The following picture illustrates all possibilities contributing to  $q_6 = 14$ :



Show that  $q_n$  is equal to the number of correctly matched strings of opening and closing brackets of length  $2n - 4$  by finding a bijection.

3. For a positive integer  $m$ , let  $\mathcal{S}_m$  be the combinatorial class of strings composed a single letter such that the length of the string is divisible by  $m$  (with the size being defined as the length of the string). Find an explicit formula for the generating function of  $\mathcal{S}_m$ .
4. For a non-negative integer  $n$ , let  $p_n$  be the number of ways one can pay  $n$  CZK using (any number of) 1, 2, and 5 CZK coins. Show that  $p_0, p_1, p_2, \dots$  is equal to the counting sequence of  $\mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_5$ , and use this observation to find an explicit formula for the generating function  $P(x)$  of this sequence. What is the radius of convergence of  $P(x)$ , and what bound does this give for the sequence  $p_0, p_1, \dots$ ?
5. Show that there exist complex numbers  $r_1, \dots, r_k$  such that  $|r_1| = \dots = |r_k| = 1$ , positive integers  $d_1, \dots, d_k \leq 2$ , and polynomials  $p, p_1, \dots, p_k$  such that

$$P(x) = \frac{p(x)}{(1-x)^3} + \sum_{i=1}^k \frac{p_i(x)}{(1-r_i x)^{d_i}}.$$

Determine the value  $p(1)$  (hint: multiply this expression by  $(1-x)^3$ , and simplify the left-hand side using the explicit formula for  $P(x)$ ).

6. In the following lecture, we will see that for every positive integer  $k$ ,  $(1 - x)^{-k}$  is the generating function of a sequence  $q_0, q_1, q_2, \dots$  such that  $q_n = \frac{n^{k-1}}{(k-1)!} + O(n^{k-2})$ . Use this fact to show that  $p_n = \frac{n^2}{20} + O(n)$ .

## Lesson 3

# Generalized binomial formula. Estimates from generating functions. Entropy and binomial coefficient estimates.

In the previous lecture, we have considered the class  $\mathcal{C}$  of well-matched strings of opening and closing brackets (with the size of the the string defined as the number of opening brackets), and we have shown that this class has the generating function

$$C(x) = \frac{1 - \sqrt{1 - 4x}}{2x}.$$

How to get an exact formula from this? We need to know something about  $\sqrt{1 - 4x} = (1 - 4x)^{1/2}$ .

### 3.1 Generalized binomial formula

We know that for a non-negative integer  $k$ , we have  $(1 + x)^k = \sum_{n=0}^k \binom{k}{n} x^n$ . We need to generalize this formula to real values of  $k$ . Let us start with the binomial coefficient. We have

$$\binom{a}{b} = \frac{a \cdot (a - 1) \cdots (a - b + 1)}{b!},$$

and there is no issue with taking this as a definition even when  $a$  is an arbitrary real number (though we still require  $b$  to be a non-negative integer).

**Example 14.** For  $n \geq 1$ , we have

$$\begin{aligned}
\binom{\frac{1}{2}}{n} &= \frac{\frac{1}{2} \cdot \left(\frac{1}{2} - 1\right) \cdots \left(\frac{1}{2} - n + 1\right)}{n!} \\
&= \frac{1}{2^n} \cdot \frac{(-1) \cdot (-3) \cdots (-(2n-3))}{n!} = \frac{(-1)^{n-1}}{2^n} \cdot \frac{1 \cdot 3 \cdots (2n-3)}{n!} \\
&= \frac{(-1)^{n-1}}{2^n} \cdot \frac{1 \cdot 3 \cdots (2n-3)}{n!} \cdot \frac{2 \cdot 4 \cdots (2n-2)}{2 \cdot 4 \cdots (2n-2)} \\
&= \frac{(-1)^{n-1}}{2^{2n-1}} \cdot \frac{(2n-2)!}{n!(n-1)!} = \frac{(-1)^{n-1}}{2^{2n-1}n} \cdot \frac{(2n-2)!}{(n-1)!(n-1)!} \\
&= \frac{(-1)^{n-1}}{2^{2n-1}n} \cdot \binom{2n-2}{n-1}.
\end{aligned}$$

Note also that  $\binom{\frac{1}{2}}{0} = 1$ .

Using e.g. the theory of Taylor series, it is easy to see that the binomial formula has the following generalization.

**Theorem** (Generalized Binomial Formula). For every real number  $a$  and for any real  $x$  such that  $|x| < 1$ ,

$$(1+x)^a = \sum_{n=0}^{\infty} \binom{a}{n} x^n.$$

Thus, when  $|x| < 1/4$ , we have

$$\begin{aligned}
(1-4x)^{1/2} &= \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} (-4x)^n \\
&= 1 + \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{2^{2n-1}n} \cdot \binom{2n-2}{n-1} \cdot (-1)^n 2^{2n} x^n \\
&= 1 - \sum_{n=1}^{\infty} \frac{2}{n} \cdot \binom{2n-2}{n-1} \cdot x^n
\end{aligned}$$

Going back to the generating function from Example 13, we have

$$\begin{aligned}
C(x) &= \frac{1 - (1-4x)^{1/2}}{2x} = \frac{\sum_{n=1}^{\infty} \frac{2}{n} \cdot \binom{2n-2}{n-1} \cdot x^n}{2x} \\
&= \sum_{n=1}^{\infty} \frac{1}{n} \cdot \binom{2n-2}{n-1} \cdot x^{n-1} = \sum_{n=0}^{\infty} \frac{1}{n+1} \cdot \binom{2n}{n} \cdot x^n.
\end{aligned}$$

It follows that the number of strings of opening and closing brackets of length  $2n$  that are correctly matched is

$$c_n = \frac{1}{n+1} \cdot \binom{2n}{n}.$$

Note that there is a natural probabilistic interpretation for this result: Suppose that we choose  $n$  out of  $2n$  positions in a string at random, put opening brackets on these positions, and closing brackets on the remaining ones. Then the probability that the resulting string is correctly matched is exactly  $\frac{1}{n+1}$ . This probability goes to 0 as  $n$  goes to infinity, but perhaps slower than you might guess.

The quantity  $c_n = \frac{1}{n+1} \cdot \binom{2n}{n}$  appears quite often in combinatorics. For example, in the tutorials and the homework, we have established bijections showing that:

- $c_{n-1}$  is equal to the number of non-decreasing sequences of integers with the first term 1, the last term  $n \geq 1$ , and each term at least as large as its position in the sequence.
- For  $n \geq 3$ , the number of ways how we can cut a convex  $n$ -gon into triangles by  $n - 3$  non-crossing line segments is equal to  $c_{n-2}$ .
- The number of binary trees with  $n$  vertices is equal to  $c_n$ .

The numbers  $c_0, c_1, c_2, \dots$  are known as *Catalan numbers*.

## 3.2 Estimates of generating function coefficients

One of the most important features of the generating function approach is that once we know a formula  $A(x)$  for the generating function  $\sum_{n=0}^{\infty} a_n x^n$  of a combinatorial class, there are many results from mathematical analysis that give asymptotic bounds on  $a_n$ . We have already seen the basic one: Determine the radius  $R$  of convergence of this series ( $R$  is typically the infimum of the values of  $x$  for which  $A(x)$  is not defined; for a precise statement, see the last part of the Safety Theorem). Then for every  $\varepsilon > 0$ , we have

$$a_n \leq (1/R + \varepsilon)^n$$

for all sufficiently large  $n$  (and  $a_n \geq (1/R - \varepsilon)^n$  for infinitely many values of  $n$ ). This approximation is however rather crude— $a_n$  can differ from  $(1/R)^n$  by a multiplicative factor that quickly grows with  $n$  (though it must be subexponential in  $n$ , at least for infinitely many values of  $n$ ).

**Example 15.** *The radius of convergence of the generating function  $C(x) = \frac{1 - (1 - 4x)^{1/2}}{2x}$  of Catalan numbers is  $1/4$ , giving us a rough estimate  $c_n \approx 4^n$ . We have also determined the exact value*

$$c_n = \frac{1}{n+1} \cdot \binom{2n}{n} = \Theta\left(\frac{4^n}{n^{3/2}}\right),$$

*which differs by the factor  $\Theta(n^{3/2})$ . This factor grows to infinity, and thus the rough estimate based on the radius of convergence becomes less and less precise.*

There are much more powerful complex-analytic methods based on the analysis of the behavior of the generating functions around the radius of convergence, which make it possible to obtain better estimates (often even asymptotically precise ones). They are however beyond the scope of this lecture (see the class “Analytic combinatorics” if you are interested).

Instead, let us show another simple bound, which works well especially for series with quickly decreasing coefficients.

**Lemma 16.** *Let  $A(x) = \sum_{n=0}^{\infty} a_n x^n$  be a series with radius of convergence  $R > 0$ , such that  $a_n \geq 0$  for every  $n$ . Then*

$$a_n \leq \inf_{0 < x < R} \frac{A(x)}{x^n}$$

holds for every non-negative integer  $n$ .

*Proof.* Let  $F(x) = \frac{A(x)}{x^n}$ , and note that

$$F(x) = \frac{a_0}{x^n} + \frac{a_1}{x^{n-1}} + \dots + \frac{a_{n-1}}{x} + a_n + \sum_{k \geq 0} a_{n+k} x^k \geq a_n$$

holds for every  $x \in (0, R)$ . Thus, we also have  $a_n \leq \inf_{0 < x < R} F(x)$ , as required.  $\square$

**Example 17.** *Consider the well-known series*

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

*This series has infinite radius of convergence (by the Safety Theorem, or since  $n!$  grows faster than any exponential function). Thus, Lemma 16 gives*

$$\frac{1}{n!} \leq \inf_{x > 0} \frac{e^x}{x^n} = \frac{e^n}{n^n};$$

*the fact that the infimum is achieved for  $x = n$  follows by considering the derivative*

$$\frac{e^x}{x^n} - \frac{ne^x}{x^{n+1}}$$

*of the function  $e^x/x^n$ . Thus,*

$$n! \geq \left(\frac{n}{e}\right)^n.$$

*This differs from the asymptotically precise Stirling’s formula*

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

*only by the factor of  $\sqrt{2\pi n}$ .*

Let us remark that the following simple upper bound

$$n! \leq en \left(\frac{n}{e}\right)^n$$

is often useful. To prove this, note that

$$\begin{aligned} \log(n-1)! &= \log 1 + \log 2 + \log 3 + \dots + \log(n-1) \\ &\leq \int_1^n \log x \, dx = [x \log x - x]_1^n \\ &= n \log n - n + 1. \end{aligned}$$

Hence,  $(n-1)! \leq e \left(\frac{n}{e}\right)^n$  and

$$n! = n(n-1)! \leq en \left(\frac{n}{e}\right)^n.$$

Again, complex analysis gives us tools to significantly improve the rough estimate from Lemma 16.

### 3.3 Entropy and binomial coefficients

It is quite often useful to be able to estimate the binomial coefficients rather precisely. It turns out that there is a quite tight connection between them and entropy. We are going to discuss the meaning of entropy in the information theory in the following lecture; for now, let us just define the *entropy function* for  $0 < p < 1$  as

$$H(p) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p};$$

additionally, we let  $H(0) = H(1) = 0$  (this extends  $H(p)$  continuously).

**Theorem 18.** *Let  $n$  be a positive integer  $n$  and let  $k \leq n/2$  be a non-negative integer. We have*

$$\frac{1}{n+1} 2^{nH(k/n)} \leq \binom{n}{k} \leq \sum_{i=0}^k \binom{n}{i} \leq 2^{nH(k/n)}.$$

*Proof.* Let  $p = k/n \leq 1/2$ . Since  $p \leq 1-p$ , we have  $p^k(1-p)^{n-k} \leq p^i(1-p)^{n-i}$  for all  $i \leq k$ . By the binomial theorem, we have

$$\begin{aligned} 1 &= (p + (1-p))^n = \sum_{i=0}^n \binom{n}{i} p^i (1-p)^{n-i} \\ &\geq \sum_{i=0}^k \binom{n}{i} p^i (1-p)^{n-i} \\ &\geq p^k (1-p)^{n-k} \sum_{i=0}^k \binom{n}{i}. \end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{i=0}^k \binom{n}{i} &\leq \frac{1}{p^k(1-p)^{n-k}} = \left( \frac{1}{p^{k/n}(1-p)^{1-k/n}} \right)^n \\
&= \left( \frac{1}{p^p(1-p)^{1-p}} \right)^n = \left( \left( \frac{1}{p} \right)^p \left( \frac{1}{1-p} \right)^{1-p} \right)^n \\
&= 2^{n(p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p})} = 2^{nH(p)} = 2^{nH(k/n)}.
\end{aligned}$$

To prove the lower bound, we need to observe that

$$\binom{n}{k} p^k (1-p)^{n-k} \geq \binom{n}{i} p^i (1-p)^{n-i}$$

holds for every  $i$  (we skip the straightforward computation). Thus,  $\binom{n}{k} p^k (1-p)^{n-k}$  is the largest term of the sum  $\sum_{i=0}^n \binom{n}{i} p^i (1-p)^{n-i} = 1$ , and consequently

$$\binom{n}{k} p^k (1-p)^{n-k} \geq \frac{1}{n+1}.$$

The bound  $\binom{n}{k} \geq \frac{1}{n+1} 2^{nH(k/n)}$  then follows by a simplification of this expression analogous to the one we performed for the upper bound.  $\square$

Using this Theorem, we can obtain a weaker but often easier to use bound. Starting with the extremely useful inequality  $1+x \leq e^x = 2^{x \log_2 e}$  valid for every real number  $x$ , we get  $\log_2(1+x) \leq x \log_2 e$ . This gives us the following upper bound on  $H(p)$ .

$$\begin{aligned}
H(p) &= p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p} = p \log_2 \frac{1}{p} + (1-p) \log_2 \left( 1 + \frac{p}{1-p} \right) \\
&\leq p \log_2 \frac{1}{p} + (1-p) \cdot \frac{p}{1-p} \log_2 e = p \log_2 \frac{1}{p} + p \log_2 e \\
&= p \log_2 \frac{e}{p}.
\end{aligned}$$

Thus,

$$\binom{n}{k} \leq \sum_{i=0}^k \binom{n}{i} \leq 2^{nH(k/n)} \leq 2^{n \cdot \frac{k}{n} \log_2 \frac{en}{k}} = 2^{k \log_2 \frac{en}{k}} = \left( \frac{en}{k} \right)^k$$

holds for all  $k \leq n/2$ .

### 3.4 Homework

1. Show that

$$\frac{1}{\sqrt{1-x}} = \sum_{n=0}^{\infty} \frac{1}{2^{2n}} \binom{2n}{n} x^n$$

holds for every real number  $x$  such that  $|x| < 1$ .

2. Let  $b_n$  be the number of ways how to divide the set  $\{1, \dots, n\}$  into (any number of) non-empty parts. For instance,  $b_3 = 5$  counts the partitions

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

It can be shown that

$$\sum_{n=0}^{\infty} \frac{b_n}{n!} \cdot x^n = e^{e^x - 1}$$

holds for every real number  $x$ . Using this fact, prove that for every non-negative integer  $n$ , we have

$$b_n \leq n \cdot \left( e^{\frac{1}{l(n)} + l(n) - 1} \right)^n,$$

where  $l(n)$  is the unique real number such that  $l(n)e^{l(n)} = n$ . If you cannot quite get to this exact expression, you can submit a solution leading to any other reasonable upper bound.

3. Show that

$$\binom{ck}{k} \leq \left( \frac{c^c}{(c-1)^{c-1}} \right)^k$$

holds for all integers  $c \geq 2$  and  $k \geq 0$ .

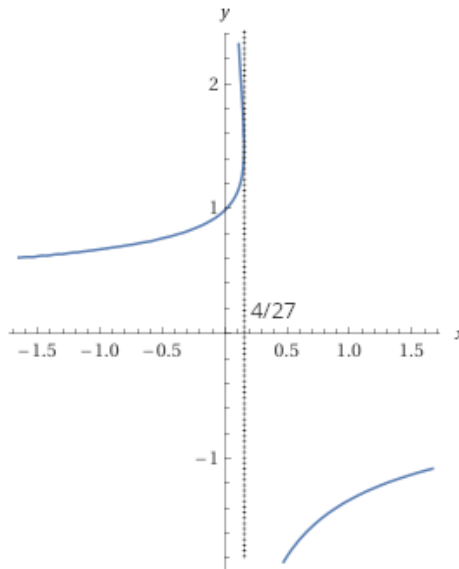
### 3.5 Tutorial

1. Show that

$$\binom{-n}{k} = (-1)^k \binom{k+n-1}{n-1}$$

holds for all integers  $n \geq 1$  and  $k \geq 0$ .

2. Suppose that the sequence  $d_0, d_1, d_2, d_3, \dots$ , where  $d_0 = 1$ ,  $d_1 = 5$ , and  $d_2 = 17$ , satisfies the recurrence  $d_n = 7d_{n-1} - 16d_{n-2} + 12d_{n-3}$  for  $n \geq 3$ . Find the generating function of this sequence and use it to determine an explicit formula for  $d_n$ .
3. Ternary trees are recursively defined as follows: A *ternary tree* either is the empty tree `nil`, or consists of the root vertex, the left son, the middle son, and the right son, where all sons are also ternary trees. The number of vertices of `nil` is 0, and the number of vertices of a non-empty ternary tree with left son  $l$ , middle son  $m$ , and right son  $r$  is  $1 + n_l + n_m + n_r$ , where  $n_l$ ,  $n_m$ , and  $n_r$  are the numbers of vertices of  $l$ ,  $m$ , and  $r$ , respectively. Let  $t_n$  be the number of ternary trees with  $n$  vertices and let  $T(x)$  be the generating function for this sequence. Show that  $T(x) = 1 + xT^3(x)$ .
4. Here is the plot of points  $(x, y)$  such that  $y = 1 + xy^3$ .



Use the information from this plot to determine the radius of convergence of  $T(x)$ . What does this tell you about  $t_n$ ?

5. Suppose  $A(x) = \sum_{n \geq 0} a_n x^n$  has non-zero but finite radius of convergence  $R$ , and that  $a_n \geq 0$  for all  $n$ . Show that for  $n \geq 10$ , we have

$$a_n \leq 3A(R \cdot (1 - 1/n)) \cdot (1/R)^n$$

(it can be useful to know that for  $n \geq 10$ , we have  $(1 - 1/n)^n \geq 1/3$ ).  
What upper bound does this give for  $t_n$ ?

6. Using a different approach, it is actually possible to show that

$$t_n = \frac{1}{n} \binom{3n}{n-1}$$

for every  $n \geq 1$ . What bounds on  $t_n$  does the entropy estimate of the binomial coefficients give?

**Part II**

**Coding theory**

## Lesson 4

# Information and entropy

We now move on to another topic, an introduction to coding theory; as we are going to see, this topic is quite related to combinatorial estimates.

Let us start with a somewhat philosophical question: How much information is there in an English text of a given length  $n$ ? For a simplification, suppose that we only consider texts consisting of lowercase letters and spaces. Very naively, the total number of strings of length  $n$  consisting of these characters is  $27^n$ , and thus to directly store such a string we need

$$\log_2 27^n = n \log_2 27 \approx 4.75n$$

bits. However, vast majority of these strings will be meaningless jumbles of characters, so this is definitely not a good measure of the information content of a text.

So, perhaps instead of counting the number of all strings, we should only count the number of valid English sentences consisting of  $n$  characters and take a logarithm of that number? Of course, it is difficult to define what a *valid English sentence* actually means; but beyond that, we run into another issue with this approach. Consider the statement “vampires eat potatoes”. This is a perfectly valid and understandable English statement, yet you are exceedingly unlikely to find it anywhere (except for these lecture notes). It should be obvious that a vast majority of syntactically and semantically valid English sentences have only extremely small probability of ever being used. So, our second attempt at the measure of information content would basically only take into account these extremely unlikely sentences, which is clearly not desirable.

On the other hand, it would be strange to completely ignore the “possible but unlikely” statements, in part because it is not at all clear what the cut-off probability should be. A way around this issue is by considering how well we could *compress* the information—we can represent the common sentences by short strings of bits, and only use long strings to represent the unlikely ones, thus saving on space on average.

Let us now develop these ideas more formally. A *message space* is a finite probability space  $M$ ; in the discussed example, the elements of  $M$  (*messages*)

would be English sentences of given length  $n$  and the probability  $\Pr[m]$  of a sentence  $m \in M$  would be the likelihood of this sentence. A *binary encoding* is an injective function from  $M$  to the set of finite sequences of 0's and 1's. For a (binary) string  $s$ , let  $|s|$  denote its length (number of characters/bits). The *average length* of an encoding  $f$  is

$$E_{m \in M} [|f(m)|] = \sum_{m \in M} \Pr[m] \cdot |f(m)|.$$

**Example 19.** Consider the message space  $M = \{a, b, c\}$  where  $\Pr[a] = 2/3$  and  $\Pr[b] = \Pr[c] = 1/6$ . The following table shows some possible encodings and their average lengths:

	$a$	$b$	$c$	avg. length
Encoding 1	0	1	00	7/6
Encoding 2	00	01	10	2
Encoding 3	0	10	11	4/3

We could now consider defining the information content to be the minimum possible average length over all encodings. However, there is still a subtlety to consider. Suppose that we e.g. use the encoding to store a message on a hard drive; thus, starting at some fixed point, we write the sequence  $s = f(m)$  of 0's and 1's on the disk. However, the disk generally does not end at the end of the sequence; rather, after it, we will have whatever 0's and 1's were written there before. How do we recognize where the encoded message  $s$  ends? We could of course separately store the length of the encoded message; however, that also requires some space (roughly  $\log_2 |s|$  bits) and it would not be fair not to take this space into account when computing the length of the encoding.

To deal with this issue, we are going to consider only *prefix-free* encodings; i.e., encodings  $f$  such that for all distinct  $m_1, m_2 \in M$ , the sequence  $f(m_1)$  is not a prefix of the sequence  $f(m_2)$ . Thus, when we read the stored string from the disk, we know that the encoded message ends as soon as the initial part that we read is equal to  $f(m)$  for some  $m \in M$  (it cannot be longer, since then there would have to exist another message  $m'$  such that  $f(m)$  is a prefix of  $f(m')$ ).

**Example 20.** Encodings 2 and 3 from the previous example are prefix-free, but Encoding 1 is not, since  $f(a) = 0$  is a prefix of  $f(c) = 00$ .

As another example, we can turn any encoding into a prefix-free one by preceding each sequence by a prefix-free encoding of its length. E.g., in Encoding 1, we can prepend 0 if the encoded message has length one and 1 if it has length two, obtaining a prefix-free encoding  $f$  such that  $f(a) = 00$ ,  $f(b) = 01$ , and  $f(c) = 100$ . Of course, this increases the average length of the encoding (to  $13/6$  in this case).

It is now natural to ask how good prefix-free encodings are possible. This turns out to be related to the important notion of entropy, which plays some role in many parts of the information theory. The *entropy* of a finite probability

space  $M$  is defined as

$$H(M) = E_{m \in M} \left[ \log_2 \frac{1}{\Pr[m]} \right] = \sum_{m \in M} \Pr[m] \cdot \log_2 \frac{1}{\Pr[m]}.$$

**Example 21.** *The message space discussed in the previous examples has entropy*

$$\frac{2}{3} \cdot \log_2 \frac{3}{2} + 2 \cdot \frac{1}{6} \cdot \log_2 6 \approx 1.25.$$

*The Bernoulli probability distribution  $\text{Bernoulli}(p)$  with mean  $p$  (i.e., the probability space  $\{0, 1\}$  such that  $\Pr[1] = p$  and  $\Pr[0] = 1 - p$ ) has entropy*

$$H(\text{Bernoulli}(p)) = p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p}.$$

*This matches the function  $H(p)$  we have seen in the previous lecture in the context of binomial coefficient estimates.*

Let us start with a construction, based on the idea of the *arithmetic coding* compression algorithm.

**Theorem 22.** *For every message space  $M$ , there exists a prefix-free encoding of  $M$  of average length at most  $H(M) + 2$ .*

*Proof.* Let us divide the interval  $[0, 1)$  into disjoint left-closed intervals  $I_m$  for  $m \in M$ , where  $I_m$  has length  $\Pr[m]$ . For each  $m \in M$ , let  $(x_m, k_m)$  be a pair of non-negative integers such that

- both  $\frac{x_m}{2^{k_m}}$  and  $\frac{x_m+1}{2^{k_m}}$  are contained in  $I_m$ , and
- $k_m$  is smallest possible.

We define  $f(m)$  as  $x_m$  written in binary with exactly  $k_m$  digits, possibly with leading zeros.

**Example 23.** *For our usual message space, we can take  $I_a = [0, 2/3)$ ,  $I_b = [2/3, 5/6)$  and  $I_c = [5/6, 1)$ . Then we can take  $(x_a, k_a) = (0, 1)$ ,  $(x_b, k_b) = (11, 4)$ , and  $(x_c, k_c) = (14, 4)$ . Hence,  $f(a) = 0$ ,  $f(b) = 1011$ , and  $f(c) = 1110$ .*

First, let us argue that this encoding is prefix-free. Suppose for a contradiction that  $f(m)$  is a prefix of  $f(m')$  for distinct  $m, m' \in M$ . Then  $k_m < k_{m'}$  and  $x_{m'}$  written in binary with exactly  $k_{m'}$  digits starts with  $x_m$  written in binary with exactly  $k_m$  digits. In other words,

$$x_{m'} = 2^{k_{m'} - k_m} x_m + r,$$

where  $0 \leq r < 2^{k_{m'} - k_m}$ . Thus,

$$\begin{aligned} \frac{x_m}{2^{k_m}} &\leq \frac{x_{m'}}{2^{k_{m'}}} = \frac{2^{k_{m'} - k_m} x_m + r}{2^{k_{m'}}} = \frac{x_m}{2^{k_m}} + \frac{r}{2^{k_{m'}}} \\ &< \frac{x_m + 1}{2^{k_m}}. \end{aligned}$$

Since both  $\frac{x_m}{2^{k_m}}$  and  $\frac{x_m + 1}{2^{k_m}}$  belong to the interval  $I_m$ , so does  $\frac{x_{m'}}{2^{k_{m'}}$ . However,  $\frac{x_{m'}}{2^{k_{m'}}$  also belongs to  $I_{m'}$ , which is a contradiction since the intervals  $I_m$  and  $I_{m'}$  are disjoint.

Now, let us consider the average length of this encoding. Let

$$\ell_m = 1 + \lceil \log_2 \frac{1}{\Pr[m]} \rceil \leq 2 + \log_2 \frac{1}{\Pr[m]}.$$

Since the interval  $I_m$  is left-closed and has length  $\Pr[m] \geq 2 \cdot \frac{1}{2^{\ell_m}}$ , two consecutive integer multiples of  $\frac{1}{2^{\ell_m}}$  belong to  $I_m$ , and by the minimality of  $k_m$ , we have  $k_m \leq \ell_m$ . Thus, the average length of  $f$  is

$$\mathbb{E}_{m \in M} [|f(m)|] = \mathbb{E}_{m \in M} [k_m] \leq \mathbb{E}_{m \in M} [\ell_m] \leq 2 + \mathbb{E}_{m \in M} \left[ \log_2 \frac{1}{\Pr[m]} \right] = 2 + H(M).$$

□

Conversely, this cannot be improved much.

**Theorem 24.** *Every prefix-free encoding  $f$  of a message space  $M$  has average length at least  $H(M)$ .*

*Proof.* For  $m \in M$ , let  $\ell_m$  be the length of  $f(m)$  and let  $\ell = \max_{m \in M} \ell_m$ . Moreover, for  $m \in M$ , let  $F(m)$  be the set of all  $2^{\ell - \ell_m}$  binary strings of length exactly  $\ell$  that start with  $f(m)$ . Note that for distinct  $m, m' \in M$ , we have  $F(m) \cap F(m') = \emptyset$ : Without loss of generality, we have  $|f(m)| \leq |f(m')|$ . A string  $s \in F(m')$  starts with  $f(m')$ , and since the encoding  $f$  is prefix-free, it cannot start with  $f(m)$ ; in other words,  $s \notin F(m)$ .

There are exactly  $2^\ell$  binary strings of length  $\ell$  and each of them belongs to  $F(m)$  for at most one  $m \in M$ . Hence,

$$\sum_{m \in M} 2^{\ell - \ell_m} = \sum_{m \in M} |F(m)| \leq 2^\ell,$$

and thus

$$\sum_{m \in M} 2^{-\ell_m} \leq 1.$$

Let us remark that this lower bound on lengths of prefix-free encodings is known as Kraft-McMillan inequality.

Let  $\gamma = \frac{1}{\sum_{m \in M} 2^{-\ell_m}}$  and for  $m \in M$  let  $q_m = \gamma 2^{-\ell_m}$ , so that  $\sum_{m \in M} q_m = 1$ ; by the previous inequality, we have  $\gamma \geq 1$ . Since  $\log_2$  is a concave function, we

have

$$\begin{aligned}
0 &= \log_2 1 = \log_2 \sum_{m \in M} q_m = \log_2 \sum_{m \in M} \Pr[m] \cdot \frac{q_m}{\Pr[m]} \\
&\geq \sum_{m \in M} \Pr[m] \cdot \log_2 \frac{q_m}{\Pr[m]} \\
&= \sum_{m \in M} \Pr[m] \cdot \log_2 \frac{1}{\Pr[m]} + \sum_{m \in M} \Pr[m] \cdot \log_2 q_m \\
&= H(M) + \sum_{m \in M} \Pr[m] \cdot \log_2 q_m.
\end{aligned}$$

Therefore,

$$\begin{aligned}
H(M) &\leq - \sum_{m \in M} \Pr[m] \cdot \log_2 q_m = - \sum_{m \in M} \Pr[m] \cdot \log_2 \gamma 2^{-\ell_m} \\
&= \sum_{m \in M} \Pr[m] \cdot (\ell_m - \log_2 \gamma) \leq \sum_{m \in M} \Pr[m] \cdot \ell_m \\
&= \sum_{m \in M} \Pr[m] \cdot |f(m)|.
\end{aligned}$$

□

Based on Theorems 22 and 24, we can imagine that the entropy is the desired measure of the information content. This intuition turns out to be confirmed by its appearance in other information-theoretic contexts (e.g., when considering the amount of information that can be carried over a noisy transmission channel, or in relation to the error-correcting codes, the topic of the next lesson).

To answer our initial question, experiments show that English text of length  $n$  has entropy around  $1.5n$ , i.e., around 1.5 bits per letter. In other words, by comparing this with our initial estimate of  $4.5n$  which assumes completely independent letters, we expect an average English text to be compressible to around 1/3 of its length without losing any information.

We can now put the bound on binomial coefficients from the previous lecture in context. For integers  $k \geq 0$  and  $n \geq 1$  such that  $k \leq n/2$ , consider the following message spaces:

- the uniform message space  $M_{n,k}$  of binary strings of length  $n$  containing exactly  $k$  digits 1 (*uniform* means each such string has the same probability  $\frac{1}{\binom{n}{k}}$ ), and
- the message space  $M'_{n,k}$  of binary strings of length  $n$  where each bit is independently 1 with probability  $k/n$  and 0 with probability  $1 - k/n$  (thus, a binary string with exactly  $t$  digits 1 has probability  $(k/n)^t (1 - k/n)^{n-t}$ ).

These message spaces are of course different. However, note that a random string selected from  $M'_{n,k}$  has with high probability roughly  $k$  digits 1, and thus

they are in a vague sense “similar”. Intuitively, we might also expect them to have similar entropy. Theorem 18 from the previous lecture confirms that this is the case. Indeed, it states that

$$nH(k/n) - \log_2(n+1) \leq \log_2 \binom{n}{k} \leq nH(k/n),$$

and since

$$H(M_{n,k}) = \binom{n}{k} \cdot \frac{1}{\binom{n}{k}} \cdot \log_2 \binom{n}{k} = \log_2 \binom{n}{k},$$

we have

$$nH(k/n) - \log_2(n+1) \leq H(M_{n,k}) \leq nH(k/n).$$

On the other hand, using the linearity of expectation (see tutorials), it is easy to see that

$$H(M'_{n,k}) = n \cdot H(\text{Bernoulli}(k/n)) = nH(k/n).$$

## 4.1 Homework

1. You are sending information over a channel as a sequence of 0's and 1's. However, due to the technical limitations of the receiver, it is not possible to send two consecutive 1's (doing so could cause them to be perceived as a single 1, leading to errors). How much information can you send over such a channel in a message of length  $n$ ?

More precisely, let  $M_n$  be the message space consisting of all  $\{0, 1\}$ -strings of length  $n$  that do not contain consecutive 1's and where all the messages have the same probability. What is the entropy of this message space? Also determine the value of

$$\lim_{n \rightarrow \infty} \frac{H(M_n)}{n}$$

(the entropy per message bit).

2. At [https://en.wikipedia.org/wiki/Letter\\_frequency](https://en.wikipedia.org/wiki/Letter_frequency), you can find a table giving the frequency of letters in English texts. For simplicity, suppose that an English text is a random sequence of  $n$  letters, each chosen independently at random with the probability given by the aforementioned table. What is the entropy of such a text? You can use any means you want to obtain the numeric answer, but you need to describe in your solution how you did it.
3. You are playing the game of “guess a number” with your friend: He picks an integer between 1 and  $n$  and you want to determine the number by asking him a series of questions. The questions can only be of form “Is your number greater than  $k$ ?” for an integer  $k$ . You have played this game with him so often that you know for each  $i \in \{1, \dots, n\}$  the probability  $p_i$  that he chooses the number  $i$ ; let  $H$  be the entropy of the corresponding probability space. Show that you can win the game using at most  $H + 2$  questions on average.

## 4.2 Tutorial

1. Consider the message space  $\{a, b, c, d\}$ , where  $P[a] = 0.6$ ,  $P[b] = P[c] = 0.15$ , and  $P[d] = 0.1$ . Find a prefix-free binary encoding of this space with minimum average length. What is the entropy of this message space?
2. Let  $M_1$  and  $M_2$  be message spaces, and let  $M = M_1 \times M_2$  be the message space of messages  $(m_1, m_2)$ , where  $m_1$  is sampled from  $M_1$  and  $m_2$  is independently sampled from  $M_2$  (thus,  $\Pr[(m_1, m_2)] = \Pr[m_1] \cdot \Pr[m_2]$ ). Show that  $H(M) = H(M_1) + H(M_2)$ .
3. Consider a text consisting of  $n$  letters, where each letter is **a** with probability 30%, **b** with probability 50%, or **c** with probability 20% (and the letters are independent). What can you say about the minimum possible average length of a prefix-free binary encoding of this text?
4. You are playing the game of “guess a word” with your friend: He chooses a word, and you need to ask him a series of yes / no questions to determine which word he chose. You are very good at this game, and so for any set  $S$  of words, you can formulate a question which has positive answer exactly for the words belonging to  $S$ . You also know for each admissible word  $w$  the probability  $p_w$  that your friend will choose this word. What can you say about the smallest possible number of questions you will on average need to determine the word?
5. Let  $M$  be a message space with at least two elements such that all elements of  $M$  have non-zero probability, let  $f$  be a prefix-free binary encoding of  $M$  with minimum possible average length  $\alpha$ , and let  $\ell = \max\{|f(m)| : m \in M\}$ .
  - Show that there exist two distinct messages  $a, b \in M$  such that  $|f(a)| = |f(b)| = \ell$ ,  $f(a)$  and  $f(b)$  differ exactly in the last bit, and every  $m \in M \setminus \{a, b\}$  satisfies  $\Pr[m] \geq \Pr[a]$  and  $\Pr[m] \geq \Pr[b]$ .
  - Let  $M'$  be the message space obtained from  $M$  by replacing the messages  $a$  and  $b$  by a single message  $q$  such that  $\Pr[q] = \Pr[a] + \Pr[b]$ , and let  $\alpha'$  be the minimum possible average length of a prefix-free encoding of  $M'$ . Show that  $\alpha = \alpha' + \Pr[a] + \Pr[b]$ .
  - Use these observations to design an algorithm to find a prefix-free encoding of  $M$  with minimum possible average length.
6. For a positive integer  $k$ , let  $M_k$  be the message space containing for every  $i \in \{0, \dots, k-1\}$  exactly  $2^i$  distinct messages with probability  $\frac{1}{k2^i}$ .
  - What is the entropy of this message space?
  - Show that  $M_k$  has a (not prefix-free) encoding with average length  $(k-1)/2$ .
  - What does this tell you about the difference between average lengths of prefix-free and non-prefix-free encodings?

## Lesson 5

# Error-correcting codes: Definitions and bounds

### 5.1 Basic notation and definitions

Detection and correction of errors is another important topic in information theory. In general, the model we consider is as follows: For transmission or storage, we need to encode a message somehow (for simplicity, we are going to only consider binary encodings, i.e., encodings as sequences of 0's and 1's). However, because of errors (noise in the transmission, imperfect writing or reading devices, damage to the media, ...), the receiving side may not be able to recover the encoded message exactly. We would like to include some redundancies in the encoding to make us able to determine what the original message was (*error correction*) or at least determine that an error occurred (*error detection*).

Of course, for this to be feasible, we need to assume something about the errors (otherwise, we could receive a completely random message, and there is obviously no way how to say anything about the original message from that). People have studied many models motivated by various real-world applications, e.g.,

- each bit could have a small probability of being flipped, independently on all others; or,
- there could be dependencies (when a bit is flipped, neighboring bits have a larger probability of being flipped as well), or
- bits from the message could be lost with some probability, or
- ...

We are going to consider one of simplest and most fundamental models: For a given parameter  $t$ , at most  $t$  bits of the encoded message can be flipped (from 0 to 1 or vice versa), but not removed or added. Furthermore, unlike the

previous lecture, we are going to only consider fixed-length encodings, where each possible message must be encoded as a sequence of 0's and 1's of a given length.

More formally, by a (*binary*) *code of length  $n$* , we mean any set  $\mathcal{C}$  of strings of length  $n$  consisting of 0's and 1's.

**Example 25.** *Consider the following codes:*

- $\mathcal{C}_1 = \{000, 111\}$  is a code of length 3
- $\mathcal{C}_2 = \{x \in \mathbb{Z}_2^n : \sum_{i=1}^n x_i = 0\}$  is a code of length  $n$ .

We should imagine that each element of the code (*codeword*) is used to encode one of the possible messages. As we are interested in codes mostly from the combinatorial perspective (how good codes are possible), we are going to mostly ignore the way how the codewords are assigned to messages and vice versa (the *encoding* and *decoding* algorithms), though for practical purposes, they (and their efficiency) are of course very important. Let us remark that  $\mathcal{C}_1$  and  $\mathcal{C}_2$  both have natural encoding algorithms: For  $\mathcal{C}_1$ , we simply replace a bit  $b$  by three copies  $bbb$  of the bit. For  $\mathcal{C}_2$ , after  $n - 1$  bits of the message, we add a *parity bit* selected so that the total number of 1's in the encoded message is even.

The *size* of the code  $\mathcal{C}$  is  $|\mathcal{C}|$ , the number of its codewords. Thus, a code of size  $s$  can be used to encode  $s$  different messages. Of course, all other things being equal, we prefer the code to be as large as possible, so that it can be used to convey more information. Often, it is more convenient to consider the base-2 logarithm of the size (the *message length*) of the code, since we can use  $\mathcal{C}$  to encode binary strings of length  $\lfloor \log_2 |\mathcal{C}| \rfloor$ . The *rate* of the code is defined as the ratio between the message length and the code length. I.e., the code with rate  $r$  makes the encoded message  $1/r$  times longer than the original one.

**Example 26.**  $\mathcal{C}_1$  has size 2 (message length 1, rate  $1/3$ ) and  $\mathcal{C}_2$  has size  $2^{n-1}$  (message length  $n - 1$ , rate  $1 - 1/n$ ).

**Observation 27.** *The length of a code is always at least as large as its message length (or equivalently, a code of length  $n$  can have at most  $2^n$  codewords), and thus the rate of a code is at most 1.*

Let us consider the simple code  $\mathcal{C}_1$ .

- How many errors (bit flips) can occur for us to be able to recover the original message? Clearly no more than one—if we are allowed to flip two bits, then we can transform both 000 and 111 to 010 and there is no way to tell from this how the original message looked like. However, we can correct one error (if the perturbed message has at least two 0's, it was originally 000, otherwise it was originally 111).
- Similarly, it is easy to see that we can detect up to two errors, since we would need to flip three bits to turn 000 to 111 or vice versa.

Based on this example, it is clear that an important parameter of the code will be the minimum number of flips needed to change one codeword into another one. For two strings  $x$  and  $y$  of the same length, the *Hamming distance*  $d(x, y)$  between  $x$  and  $y$  is the number of coordinates  $i \in \{1, \dots, n\}$  such that  $x_i \neq y_i$ .

**Example 28.**

$$d(001011, 000110) = 3$$

The (*minimum*) *distance* of a code  $\mathcal{C}$  is the minimum Hamming distance between its codewords, i.e.,  $\min\{d(x, y) : x, y \in \mathcal{C}, x \neq y\}$ .

**Example 29.**  $\mathcal{C}_1$  has distance 3,  $\mathcal{C}_2$  has distance 2.

**Observation 30.** Let  $\mathcal{C}$  be a code and  $k \geq 0$  an integer.

- If  $\mathcal{C}$  has distance at least  $2k + 1$ , then we can use  $\mathcal{C}$  to correct up to  $k$  errors.
- If  $\mathcal{C}$  has distance at least  $k + 1$ , then we can use  $\mathcal{C}$  to detect up to  $k$  errors.

*Proof.* For the first claim, each binary string  $x$  (the codeword with up to  $k$  errors) can be at Hamming distance at most  $k$  from at most one codeword from  $\mathcal{C}$ . Indeed, if  $d(x, y_1) \leq k$  and  $d(x, y_2) \leq k$  for  $y_1, y_2 \in \mathcal{C}$ , then  $d(y_1, y_2) \leq d(x, y_1) + d(x, y_2) \leq 2k$ , and since  $\mathcal{C}$  has minimum distance greater than  $2k$ , we must have  $y_1 = y_2$ .

For the second claim, note that in this situation, it is not possible to change a codeword into another valid codeword by up to  $k$  bit flips.  $\square$

All else being equal, it is thus better for a code to have as large minimum distance as possible. A related parameter of the code is its *relative distance*, the ratio of the distance and the length of the code. Thus, relative distance expresses how large fraction of the encoded message can be corrupted before we are unable to detect that an error occurred.

**Example 31.**  $\mathcal{C}_1$  has relative distance 1 and  $\mathcal{C}_2$  has relative distance  $2/n$ .

To make it easier to speak about the codes, we say that a code  $\mathcal{C}$  is an  $(n, k, d)$ -code if it has length  $n$ , message length  $k$ , and distance at least  $d$ .

**Example 32.**  $\mathcal{C}_1$  is a  $(3, 1, 3)$ -code, and  $\mathcal{C}_2$  is an  $(n, n - 1, 2)$ -code.

Let us remark that in the “ $(n, k, d)$ -code” notation, we require that the distance is at least  $d$ , rather than exactly  $d$ . Indeed, because of Observation 30, it is more important to be able to bound the distance of the code from below; and requiring equality would make the notation more cumbersome to use. The two goals of maximizing the rate and the relative distance of a code are of course in conflict. Let us now give two fundamental inequalities that all codes must obey.

## 5.2 Singleton bound

The *truncation* of a code  $\mathcal{C}$  is the code  $\mathcal{C}'$  obtained from  $\mathcal{C}$  by removing the last bit of each codeword.

**Observation 33.** *If  $\mathcal{C}$  is an  $(n, k, d)$ -code and  $d \geq 2$ , then the truncation of  $\mathcal{C}$  is an  $(n - 1, k, d - 1)$ -code.*

**Corollary 34** (Singleton bound). *Let  $n \geq d$  be positive integers and let  $k$  be a non-negative real number. If there exists an  $(n, k, d)$ -code, then  $k \leq n - d + 1$ .*

*Proof.* If  $\mathcal{C}$  is an  $(n, k, d)$ -code, then by truncating it  $d - 1$  times, we obtain an  $(n - d + 1, k, 1)$ -code. By Observation 27, it follows that  $k \leq n - d + 1$ .  $\square$

By dividing both sides of this inequality by  $n$ , we obtain the following asymptotic version of this inequality.

**Corollary 35.** *For every  $\varepsilon > 0$  and  $\delta \in [0, 1]$ , there exists  $n_0$  such that every code of length at least  $n \geq 0$  and relative distance  $\delta$  has rate at most  $1 - \delta + \varepsilon$ .*

## 5.3 Hamming bound

For a binary string  $s$  and a non-negative integer  $m$ , let  $B(s, m)$  denote the set of binary strings that can be obtained from  $s$  by at most  $m$  flips; thus,  $B(s, m)$  is the ball of radius  $m$  around  $s$  in the Hamming distance.

**Observation 36.** *Let  $\mathcal{C}$  be a code and let  $m$  be a non-negative integer such that the distance of  $\mathcal{C}$  is greater than  $2m$ . Then for any distinct codewords  $s_1, s_2 \in \mathcal{C}$ , the balls  $B(s_1, m)$  and  $B(s_2, m)$  are disjoint.*

Thus, for a code  $\mathcal{C}$  of distance  $d$ , the balls of radius  $\lfloor (d - 1)/2 \rfloor$  around the codewords have to all fit disjointly. Note that the number of elements each balls depends only on the length  $n$  of the code and the radius (but not the string it is centered on); specifically, it is equal to

$$\binom{n}{\leq r} = \sum_{i=0}^r \binom{n}{i}.$$

Indeed, this is the number of ways how we can choose at most  $r$  of the  $n$  digits of the string to flip.

**Corollary 37** (Hamming bound). *Let  $n \geq d$  be positive integers and let  $k$  be a non-negative real number. If there exists an  $(n, k, d)$ -code, then*

$$k \leq n - \log_2 \binom{n}{\leq \lfloor (d - 1)/2 \rfloor}.$$

*Proof.* The maximum number of Hamming balls of radius  $\lfloor (d-1)/2 \rfloor$  that fit disjointly to the space  $\mathbb{Z}_2^n$  of possible codewords is at most

$$\frac{2^n}{\binom{n}{\leq \lfloor (d-1)/2 \rfloor}},$$

and the desired inequality follows by taking the logarithm.  $\square$

Again, we can divide both sides of the Hamming bound inequality by  $n$  and asymptotic version. To simplify it, we use the bound

$$\log_2 \binom{n}{\leq r} \geq \log_2 \binom{n}{r} \geq nH(r/n) - \log_2(n+1)$$

from the third lecture, valid for  $r \leq n/2$ .

**Corollary 38.** *For every  $\varepsilon > 0$  and  $\delta \in [0, 1]$ , there exists  $n_0$  such that every code of length at least  $n \geq 0$  and relative distance  $\delta$  has rate at most  $1 - H(\delta/2) + \varepsilon$ .*

Thus, we see another appearance of entropy in the context of information theory.

## 5.4 A lower bound – in tutorials

How close to the Hamming bound can one get? A natural way of obtaining a large code of given length  $n$  and distance (at least)  $d$  is to simply pick codewords at distance at least  $d$  from one another as long as possible; we call such a code a *distance- $d$  greedy code of length  $n$* .

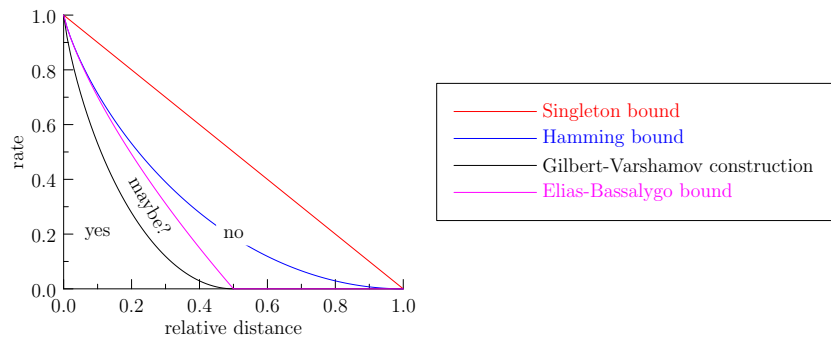
**Lemma 39.** *For all positive integers  $n$  and  $d$ , each distance- $d$  greedy code  $\mathcal{C}$  of length  $n$  has message length at least  $n - \log_2 \binom{n}{\leq d-1}$ .*

*Proof.* When we pick a codeword  $s$  to add to  $\mathcal{C}$ , this prevents us from adding exactly the  $\binom{n}{\leq d-1}$  codewords in the Hamming ball  $B(s, d-1)$  (some of which could have already been excluded because of other previously picked codewords). Thus, we can definitely pick a codeword to add to  $\mathcal{C}$  at least  $2^n / \binom{n}{\leq d-1}$  times, and the desired inequality follows by taking the logarithm.  $\square$

Again, it is natural to express this in the asymptotic way.

**Corollary 40.** *For every  $\varepsilon > 0$  and  $\delta \in [0, 0.5]$ , there exists  $n_0$  such that for every  $n \geq n_0$ , there exists a code of length  $n$ , relative distance at least  $\delta$ , and rate at least  $1 - H(\delta) - \varepsilon$ .*

Let us note that this construction is by Gilbert and independently by Varshamov. In summary, we get the following picture for the possible combinations of relative distance and rate. The picture also includes another (stronger) bound by Elias and Bassalygo in the picture.

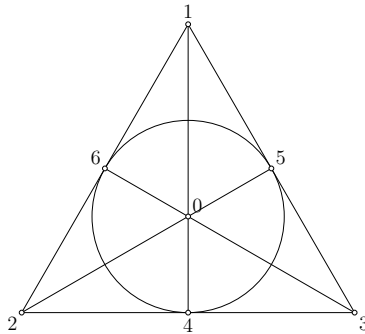


## 5.5 Homework

1. Let  $\mathcal{C}_1$  be an  $(n_1, k_1, d_1)$ -code and  $\mathcal{C}_2$  an  $(n_2, k_2, d_2)$ -code. Let  $\mathcal{C} = \{w_1w_2 : w_1 \in \mathcal{C}_1, w_2 \in \mathcal{C}_2\}$  be the code whose elements are concatenations of the elements of  $\mathcal{C}_1$  and  $\mathcal{C}_2$ . What are the parameters (length, message length, distance) of  $\mathcal{C}$ ?
2. You are playing the game of “guess a word” with your friend: He chooses one of  $m$  possible words, and you can ask him a series of  $n$  yes / no questions to determine which word he chose. You are very good at this game, and so for any set  $W$  of words, you can formulate a question which has positive answer exactly for the words belonging to  $W$ . Your friend is honest and answers the questions correctly, but up to  $r$  times, he can refuse to answer the question. Show that if there exists an  $(n, k, r + 1)$ -code such that  $m \leq 2^k$ , then you have a strategy that allows you to win every time.
3. The *Fano plane*  $F$  is the set

$$\{\{1, 2, 6\}, \{2, 3, 4\}, \{1, 3, 5\}, \{0, 1, 4\}, \{0, 2, 5\}, \{0, 3, 6\}, \{4, 5, 6\}\}$$

(graphically, the elements of  $F$  correspond to the sets of points joined by a straight line or a circle in the following picture):



This set has the property that any distinct  $x, y \in F$  intersect in exactly one element. Let  $\mathcal{C}$  be the set containing

- the strings 0000000 and 1111111, and
- for each  $x \in F$ , the binary strings  $o_x$  and  $z_x$  of length 7, where  $o_x$  has 1's exactly at positions corresponding to the elements of  $x$  and  $z_x$  has 0's exactly at these positions. So, for example, for  $x = \{1, 2, 6\}$ , we have  $o_x = 0110001$  and  $z_x = 1001110$ .

Determine the parameters (length, message length, distance) of the code  $\mathcal{C}$ .

## 5.6 Tutorial

1. Let  $S$  be a set of strings (not necessarily binary ones, i.e., the letters appearing in them can be arbitrary). Show that the Hamming distance is a metric on  $S$ .
2. A code  $\mathcal{C}$  of length  $n$  is *distance- $d$  maximal* if it has distance at least  $d$  and every string in  $\{0, 1\}^n$  is at Hamming distance less than  $d$  from a codeword of  $\mathcal{C}$ . That is,  $\mathcal{C}$  is any code that cannot be enlarged while preserving its distance. Show that  $\mathcal{C}$  has message length at least  $n - \log_2 \binom{n}{\leq d-1}$  and that if  $d \leq n/2$ , then the rate of this code is at least  $1 - H(d/n)$ .
3. For a binary string  $w$ , let  $\bar{w}$  be the string obtained from  $w$  by exchanging 0's and 1's. Let  $H_1 = \{00, 01\}$  and for  $t \geq 2$ , let  $H_t = \{w\bar{w}, w\bar{w} : w \in H_{t-1}\}$ . Determine the length, size, message length, rate, distance, and relative distance of the code  $H_t$ . Hint: Once you guess what the distance  $d_t$  of  $H_t$  is, show by induction on  $t$  that  $d(w_1, w_2) \geq d_t$  and  $d(w_1, \bar{w}_2) \geq d_t$  holds for all distinct  $w_1, w_2 \in H_t$ .
4. Let  $\mathcal{C}$  be an  $(n, k, d)$ -code, where  $d$  is odd. Let  $\mathcal{C}'$  be the code obtained from  $\mathcal{C}$  by appending to each string  $w$  in  $\mathcal{C}$  the parity bit, i.e., the bit 1 if  $w$  contains odd number of 1's and the bit 0 otherwise. Show that  $\mathcal{C}'$  is a  $(n+1, k, d+1)$ -code.
5. You wrote data to a disk using a code of length  $n$  and distance  $d$ . After some time, you tried to read the data, but realized that a part of the disk failed and you cannot read bits whose indices belong to a set  $B \subseteq \{1, \dots, n\}$  (you know this set). Show that if  $|B| \leq d-1$ , then you can still recover the original message.
6. You are playing the game of “guess a word” with your friend: He chooses one of  $m$  possible words, and you can ask him a series of  $n$  yes / no questions to determine which word he chose. You are very good at this game, and so for any set  $W$  of words, you can formulate a question which has positive answer exactly for the words belonging to  $W$ . However, your friend can lie up to  $\ell$  times. Prove that following claims are equivalent:
  - There exists a strategy that allows you to win every time.
  - There exists an  $(n, k, 2\ell + 1)$ -code such that  $m \leq 2^k$ .
7. Use the bounds from the lecture (and from the second exercise) to obtain (as good as possible) statements of the following form:
  - If  $\ell \leq \dots$  (a bound depending on  $n$  and  $m$ ), then there definitely exists a winning strategy for the game from the previous task.
  - If  $\ell \geq \dots$ , then there definitely does not exist a winning strategy.

## Lesson 6

# Linear codes. Hamming and Reed-Solomon codes.

### 6.1 Linear codes

For a general  $(n, k, d)$ -code  $\mathcal{C}$ , we face a number of difficulties:

- Determining the distance requires us to consider all pairs of distinct elements from  $\mathcal{C}$ .
- For encoding and decoding, there are no general approaches beyond rather inefficient “have a table assigning the codeword to each of the  $2^k$  possible messages” and “have a table mapping each of  $2^n$  binary strings to the nearest codeword” (though of course, there may be better encoding / decoding algorithms for particular codes).

For *linear codes*, these issues are simplified. The words of a binary code of length  $n$  can be viewed as  $n$ -dimensional vectors with coordinates belonging to  $\mathbb{Z}_2$ . In this view, a code  $\mathcal{C}$  of length  $n$  is *linear* if it forms a linear subspace of  $\mathbb{Z}_2^n$ . In other words, for every  $x, y \in \mathcal{C}$ , we have  $x + y \in \mathcal{C}$  (since we work over  $\mathbb{Z}_2$ , we do not have to worry about multiplication by a scalar; there are only two possible scalar factors,  $0 \cdot x = 0 = x + x$  and  $1 \cdot x = x$ ). In particular, note that this implies that the zero vector is an element of  $\mathcal{C}$ .

**Example 41.** *The codes  $\{000, 111\}$  and  $\{000, 011, 101, 110\}$  are linear, but the code  $\{000, 010, 101\}$  is not, since  $010 + 101 = 111$  is not contained in the code.*

Consider a linear code  $\mathcal{C}$ . Since  $\mathcal{C}$  is a (finite) linear space, it has a basis  $w_1, \dots, w_k$ ; that is, there exist linearly independent words  $w_1, \dots, w_k \in \mathcal{C}$ , where  $k = \dim \mathcal{C}$ , such that each element of  $\mathcal{C}$  is a unique linear combination of the basis words.

**Example 42.** *The code  $\{000, 111\}$  has basis 111, and the code  $\{000, 011, 101, 110\}$  has basis (e.g.) 011, 101.*

There are  $2^k$  linear combinations of  $k$  basis elements (each of them appears with the coefficient 0 or 1), and thus the size of the linear code  $\mathcal{C}$  is  $2^k$ , and its message length is  $\log_2 2^k = k = \dim \mathcal{C}$ . This gives a natural encoding algorithm: The message  $b_1 b_2 \dots b_k$  is encoded as the codeword

$$b_1 \cdot w_1 + \dots + b_k \cdot w_k.$$

For this, we need to only remember the basis of the code ( $kn$  bits in total) rather than having a table for all  $2^k$  possible messages.

Another way to describe a linear code is by noticing that a linear subspace can be described as a solution to the system of linear equations, that is, by listing a maximal set of linearly independent linear constraints satisfied by its vectors. More precisely, the *check matrix* of a linear code  $\mathcal{C}$  of length  $n$  is an  $r \times n$  matrix  $A$  (with elements from  $\mathbb{Z}_2$ ) with linearly independent rows such that a word  $w \in \mathbb{Z}_2^n$  belongs to  $\mathcal{C}$  if and only if  $Aw^T = 0$ .

**Example 43.**

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

is a check matrix of the code  $\{000, 111\}$ .

$$(1 \quad 1 \quad 1)$$

is a check matrix of the code  $\{000, 011, 101, 110\}$ .

Thus, check matrix gives us an easy way to test whether a word belongs to the code, i.e., to detect errors. It also simplifies error correction, as we are going to see in the tutorials. Note that each (independent) linear constraint reduces the dimension of the subspace by one, and thus the check matrix of a linear  $(n, k, d)$ -code has exactly  $n - k$  rows.

Finally, the following lemma makes it simpler to determine the distance of a linear code. The *weight*  $\omega(x)$  of a string  $x$  of digits is the number of its non-zero entries; that is,  $\omega(x) = d(x, 0)$ .

**Lemma 44.** *The distance of a linear code  $\mathcal{C}$  is equal to*

$$d_0 = \min_{x \in \mathcal{C} \setminus \{0\}} \omega(x),$$

that is, the minimum number of 1's in a non-zero codeword. Moreover, if  $A$  is a check matrix of  $\mathcal{C}$ , then  $d_0$  is equal to the minimum non-zero number of columns of  $A$  whose sum is 0.

*Proof.* Note that  $d_0$  is equal to the distance between some non-zero codeword  $x \in \mathcal{C}$  and the zero codeword (which also belongs to  $\mathcal{C}$ ), and thus the distance  $d$  of  $\mathcal{C}$  is at most  $d_0$ . Consider now distinct words  $w_1, w_2 \in \mathcal{C}$  such that  $d = d(w_1, w_2)$ . Observe that

$$d = d(w_1, w_2) = d(0, w_1 + w_2) = \omega(w_1 + w_2) \geq d_0.$$

Indeed, for each  $i$ , the  $i$ -th bits of  $w_1$  and  $w_2$  are different if and only if their sum (in  $\mathbb{Z}_2$ ) is 1, i.e., iff the  $i$ -th bit contributes one towards the weight of  $w_1 + w_2$ . Moreover,  $w_1 + w_2 \in \mathcal{C}$ , since  $\mathcal{C}$  is a linear code, and thus  $d_0$  is at most  $\omega(w_1 + w_2)$ . We have argued that  $d \leq d_0$  and  $d \geq d_0$ , and thus  $d = d_0$ .

Let us now consider a check matrix  $A$  of  $\mathcal{C}$ . Let  $n$  be the length of  $\mathcal{C}$ . Recall that codewords of  $\mathcal{C}$  are exactly the vectors  $x \in \mathbb{Z}_2^n$  such that  $Ax^T = 0$ , i.e., such that the columns of  $A$  whose indices correspond to the coordinates of 1's in  $x$  sum to 0. Thus, the minimum number of 1's in a non-zero codeword is equal to the minimum non-zero number of columns of  $A$  whose sum is 0.  $\square$

## 6.2 Hamming code and perfect codes

Let us now use Lemma 44 to construct a nice linear code of distance 3. It suffices to choose a check matrix  $A$  of the code. Let us fix the number  $r \geq 2$  of rows of this matrix; to make the code as large as possible, we want the number of columns to be as large as possible. On the other hand, by Lemma 44, we need to ensure that  $A$  satisfies the following conditions in order for the distance of the code to be (at least) three:

- No column can be equal to 0 (as then the distance of the code would be 1).
- No two columns can sum to 0 (as then the distance of the code would be at most 2). Equivalently, any two columns have to be different.

So, let the columns of  $A$  be exactly all distinct non-zero binary vectors of length  $r$ . It is easy to see that the rows of  $A$  are linearly independent, since  $A$  contains the  $r \times r$  identity matrix as a submatrix.

**Example 45.** For  $r = 2$ , we can choose

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

The corresponding code  $\{x : Ax = 0\}$  is  $\{000, 111\}$ .

Let us now consider the parameters of this code. The length is the number of columns of  $A$ , which is the number of non-zero binary vectors of dimension  $r$ , i.e.,  $2^r - 1$ . As we have argued above, the message length is equal to the number of columns minus the number of rows, that is,  $2^r - r - 1$ . Finally, we chose the check matrix so that the distance is at least three, and it is easy to find three columns that sum to 0, and thus the distance is exactly 3. Hence, for each positive integer  $r$ , we obtain a  $(2^r - 1, 2^r - r - 1, 3)$ -code. These codes are called *Hamming codes*.

Recall that in the previous lecture, we proved the following Hamming bound: Every code of length  $n$  and distance  $d$  has message length at most

$$n - \log_2 \binom{n}{\leq \lfloor (d-1)/2 \rfloor}.$$

Note that

$$\binom{n}{\leq 1} = \binom{n}{1} + \binom{n}{0} = n + 1,$$

and in particular, this tells us that a code of length  $n$  and distance 3 has message length at most  $n - \log_2(n + 1)$ . This is exactly satisfied by the Hamming codes, which have  $n = 2^r - 1$ ,  $\log_2(n + 1) = r$ , and message length  $2^r - r - 1$ .

In general, we say that an  $(n, k, d)$ -code is *perfect* if  $k = n - \log_2 \binom{n}{\leq \lfloor (d-1)/2 \rfloor}$ , i.e., if it perfectly matches the Hamming bound. Such codes are very rare; indeed, it is possible to show that the only possible parameters for a perfect code are:

- $(n, 0, \infty)$  for every  $n \in \mathbb{N}$ , the code consisting of a single word of length  $n$ .
- $(n, n, 1)$  for every  $n \in \mathbb{N}$ , the code containing all words of length  $n$ .
- $(n, 1, n)$  for every odd  $n \in \mathbb{N}$ , the code containing a single word of length  $n$  and its complement (obtained by flipping 0's and 1's).
- $(2^r - 1, 2^r - r - 1, 3)$  for every integer  $r \geq 2$ , the Hamming codes with the parameter  $r$ .
- $(23, 12, 7)$ , the *Golay code*, see the tutorials.

### 6.3 Reed-Solomon codes

With regards to the example from the beginning of the lecture, QR codes use *Reed-Solomon* code for error-correction. It is an example of a code which is not (in general) binary; rather, the codewords are strings of elements of a finite field  $\mathbb{F}$ . Their construction is based on the following well-known algebraic statement.

**Theorem 46.** *Let  $\mathbb{F}$  be a field and let  $p_1, p_2 \in \mathbb{F}(x)$  be distinct polynomials over  $\mathbb{F}$ . If  $p_1$  and  $p_2$  have degree at most  $s$ , then there exist at most  $s$  values  $a \in \mathbb{F}$  such that  $p_1(a) = p_2(a)$ .*

Indeed, we have  $p_1(x) = p_2(x)$  exactly for the roots of the polynomial  $p_1 - p_2$ , and we know that a polynomial of degree at most  $s$  has at most  $s$  distinct roots.

For positive integers  $b > s$  and a finite field  $\mathbb{F}$  of size at least  $b$ , we can construct a Reed-Solomon code as follows: Choose  $b$  distinct values  $a_1, \dots, a_b \in \mathbb{F}$ . For every polynomial  $p \in \mathbb{F}(x)$  of degree at most  $s$ , add the codeword  $(p(a_1), \dots, p(a_b))$ .

**Observation 47.** *Let  $\mathcal{C}$  be a Reed-Solomon code for parameters  $b > s$  and a finite field  $\mathbb{F}$ . Then  $|\mathcal{C}| = |\mathbb{F}|^{s+1}$  and any two codewords of  $\mathcal{C}$  differ in at least  $b - s$  coordinates.*

*Proof.* The codewords correspond to polynomials of degree at most  $s$  over  $\mathbb{F}$ , and each such polynomial is uniquely determined by its coefficients, i.e., an  $(s + 1)$ -tuple of elements of  $\mathbb{F}$ . Thus, the number of codewords is equal to  $|\mathbb{F}|^{s+1}$ . For any two codewords, the corresponding polynomials agree in at most  $s$  points, and thus the codewords differ in at least  $b - s$  coordinates.  $\square$

Typically, we need to store or transmit the information in binary. For this, the most convenient choice for the field  $\mathbb{F}$  is a finite field  $\mathbb{F}_{2^q}$  of size  $2^q$  (for a positive integer  $q$ ), whose elements can be naturally interpreted as binary strings of length  $q$ . Then, the codeword  $(p(a_1), \dots, p(a_b))$  of the Reed-Solomon code can be viewed as the concatenation of the binary strings corresponding to the values  $p(a_1), \dots, p(a_b)$ . The parameters of the corresponding code are

- length:  $bq$
- message length:  $\log_2 |\mathbb{F}|^{s+1} = (s + 1)q$
- distance:  $b - s$ .

These codes are very good at dealing with a particular type of errors: Suppose that the corrupted bits are clustered close to one another. If several of these bits fall into the same substring of length  $q$  corresponding to a single value from  $\mathbb{F}$ , this changes the original codeword (in  $\mathbb{F}^b$ ) only in a single coordinate. Thus, in this situation, we may be able to correct up to  $\lfloor (b - s)/2 \rfloor \cdot q$  errors.

## 6.4 Homework

1. Let  $\mathcal{C}$  be the linear code with basis 1000001, 0100010, 0010100, 0001111. Find a check matrix for  $\mathcal{C}$ , and determine the parameters (length, message length, distance) of this code.
2. Let  $\mathcal{C}$  be the linear code with check matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Find a basis for  $\mathcal{C}$ , and determine the parameters (length, message length, distance) of this code.

3. Let  $\mathcal{C}_1$  be a Hamming code of length  $n_1$  and let  $\mathcal{C}_2$  be a Reed-Solomon code (using a field whose size is a power of two) of length  $n_2$  and minimum distance 3, and let  $k_1$  and  $k_2$  be the message lengths of these codes. Suppose that the parameters of these codes are chosen so that  $n_1$  and  $n_2$  are almost the same, say so that  $n \leq n_1, n_2 \leq n + 100$  for a positive integer  $n$ . Show that  $k_1 \geq k_2 + \Omega(\log n)$ ; that is, there exists a constant  $c$  such that  $k_1 \geq k_2 + c \log n$  holds for sufficiently large  $n$ .

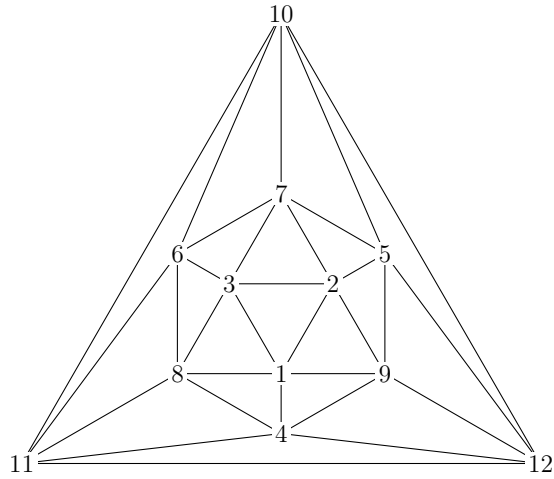
## 6.5 Tutorial

1. Let  $\mathcal{C}$  be the linear code with basis 10001010, 01001001, 00100110, 00010101. Find a check matrix for this code, and determine the parameters (length, message length, distance) of this code.
2. Let  $\mathcal{C}$  be the Hamming code of length 7 (with the parameter  $r = 3$ ). Find a basis of this code.
3. For positive integers  $d \leq n$ , let  $\mathcal{C}$  be the code of length  $n$  constructed as follows: Let  $\mathcal{C}_0$  contain only the zero vector in  $\mathbb{Z}_2^n$ . For  $i = 1, 2, \dots$ , if there exists a vector  $w_i \in \mathbb{Z}_2^n$  whose Hamming distance from all codewords of  $\mathcal{C}_{i-1}$  is at least  $d$ , then let

$$\mathcal{C}_i = \mathcal{C}_{i-1} \cup \{w + w_i : w \in \mathcal{C}_{i-1}\}.$$

Otherwise, we let  $\mathcal{C} = \mathcal{C}_{i-1}$  and the construction is finished.

- Show that  $\mathcal{C}$  is a linear code.
  - Show the distance of  $\mathcal{C}$  is at least  $d$ .
  - Show that the message length of  $\mathcal{C}$  is at least  $n - \log_2 \binom{n}{\leq d}$ .
4. Let  $A$  be the check matrix of a linear code  $\mathcal{C}$  of length  $n$  and distance  $d$ .
    - Let  $w_1, w_2 \in \mathcal{C}$  be two codewords and let  $w'_1$  and  $w'_2$  be the words obtained from them by flipping the same bits (i.e., when computing in  $\mathbb{Z}_2^n$ , we have  $w'_1 - w_1 = w'_2 - w_2$ ). Show that  $A(w'_1)^T = A(w'_2)^T$ . How does this expression depend on which bits are flipped?
    - Let  $e_1, e_2 \in \mathbb{Z}_2^n$  be distinct vectors such that  $\omega(e_1) + \omega(e_2) < d$ . Show that  $Ae_1^T \neq Ae_2^T$ .
    - Design an algorithm that for  $t < d/2$ , given a word  $w'$  obtained from a codeword  $w \in \mathcal{C}$  by flipping at most  $t$  bits, correctly determines the original word  $w$ . The algorithm is allowed to use a precomputed table containing at most  $O(n^t)$  entries.
  5. The *icosahedron* is the following graph  $G$ :



Let  $A$  be the *non-adjacency* matrix of this graph, i.e.,  $A_{i,j} = 0$  if the vertices  $i$  and  $j$  are adjacent and  $A_{i,j} = 1$  otherwise (and in particular,  $A_{i,i} = 1$  for every  $i$ ). The *extended Golay code*  $\mathcal{G}$  is the linear code (of length 24) whose check matrix is  $C = (I_{12}|A)$ , i.e., the concatenation of the  $12 \times 12$  identity matrix  $I_{12}$  with  $A$ .

- Show that  $CC^T = 0$ , and that this implies that the rows of  $C$  also form a basis of this linear code  $\mathcal{G}$ .
  - Show that this implies that  $w_1 w_2^T = 0$  for all codewords  $w_1, w_2 \in \mathcal{G}$ , and thus the number of bits which are 1 both in  $w_1$  and  $w_2$  is divisible by 2.
  - Observe that the number of 1's in each row of  $C$  is 8, and use this and the previous point to show that  $\omega(w)$  is divisible by 4 for every  $w \in \mathcal{G}$ . What does this tell you about the distance of  $\mathcal{G}$ ?
  - The distance of the extended Golay code  $\mathcal{G}$  is actually 8. What would you need to prove about the icosahedron graph to show that this is indeed true?
6. The *Golay code* is the truncation of the extended Golay code. Determine the parameters (length, message length, distance) of the Golay code and show that this code is perfect.

## Part III

# Extremal, Ramsey, and probabilistic graph theory

We are now going to consider three somewhat related aspects of graph theory. For a given property  $\pi$  (say, the property “the graph contains a triangle”):

- How many edges can a graph at most have without satisfying the property  $\pi$  (*extremal graph theory*)?
- Do all sufficiently large graphs have the property  $\pi$  (*Ramsey theory*)?
- How likely is a randomly chosen graph to satisfy the property  $\pi$  (*random graph theory*)?

## Lesson 7

# Introduction to extremal graph theory (and double-counting arguments)

### 7.1 Triangle-free graphs

What is the largest number of edges a triangle-free graph on a given number  $n$  of vertices can have? Bipartite graphs clearly do not contain triangles, and among them, the (as close to as possible) balanced complete bipartite graph is the one with largest number of edges, equal to

$$\lfloor \frac{n}{2} \rfloor \cdot \lceil \frac{n}{2} \rceil = \begin{cases} \frac{n^2}{4} & \text{if } n \text{ is even} \\ \frac{n^2-1}{4} & \text{if } n \text{ is odd} \end{cases} = \left\lfloor \frac{n^2}{4} \right\rfloor.$$

This turns out to be the best possible.

**Theorem 48** (Mantel). *Every triangle-free  $n$ -vertex graph has at most  $\frac{n^2}{4}$  edges.*

*Proof.* Let  $G$  be any  $n$ -vertex triangle-free graph and let  $e$  be the number of edges of  $G$ . Let  $m(G)$  be the number of triples  $(u, v_1, v_2)$  of vertices of  $G$  such that  $uv_1, uv_2 \in E(G)$ . Let us note that the vertices  $v_1$  and  $v_2$  do not have to be distinct (i.e.,  $v_1 = v_2$  is possible) and that the order of vertices matter (i.e., if  $y$  and  $z$  are distinct neighbors of a vertex  $x$ , then both triples  $(x, y, z)$  and  $(x, z, y)$  contribute to  $m(G)$ ).

On one hand, for each vertex  $u$ , we can choose  $v_1$  and  $v_2$  among the neighbors of  $u$  arbitrarily, and thus

$$m(G) = \sum_{u \in V(G)} \deg^2 u. \tag{7.1}$$

On the other hand, note that since  $G$  is triangle-free, if  $uv_1, uv_2 \in E(G)$ , then  $v_1v_2 \notin E(G)$  (and this is the case even if  $v_1 = v_2$ ). Thus, for each vertex  $v_1$ ,

we can choose one of its neighbors as  $u$  and one of its non-neighbors (including  $v_1$  itself) as  $v_2$ . This choice does not guarantee that  $uv_2 \in E(G)$ , and thus this can potentially overestimate  $m(G)$ ; nevertheless, we at least get the following upper bound:

$$\begin{aligned} m(G) &\leq \sum_{v_1 \in V(G)} \deg v_1 \cdot (n - \deg v_1) = n \cdot \sum_{v_1 \in V(G)} \deg v_1 - \sum_{v_1 \in V(G)} \deg^2 v_1 \\ &= 2ne - \sum_{v_1 \in V(G)} \deg^2 v_1. \end{aligned} \quad (7.2)$$

By combining (7.1) and (7.2), we get

$$\begin{aligned} 2ne - \sum_{v_1 \in V(G)} \deg^2 v_1 &\geq m(G) = \sum_{u \in V(G)} \deg^2 u \\ 2ne &\geq 2 \sum_{v \in V(G)} \deg^2 v \\ e &\geq \frac{\sum_{v \in V(G)} \deg^2 v}{n} \end{aligned} \quad (7.3)$$

The function  $f(x) = x^2$  is convex, and thus

$$\frac{\sum_{v \in V(G)} \deg^2 v}{n} \geq \left( \frac{\sum_{v \in V(G)} \deg v}{n} \right)^2 = \left( \frac{2e}{n} \right)^2. \quad (7.4)$$

Thus, (7.3) gives

$$e \geq \frac{\sum_{v \in V(G)} \deg^2 v}{n} \geq \left( \frac{2e}{n} \right)^2,$$

and consequently

$$e \leq \frac{n^2}{4}.$$

□

The proof gives an example of a *double-counting argument*, where we compute the same quantity in two different ways, obtaining the desired claim. Clearly, the tricky part is figuring out the right quantity to compute. Interestingly, researchers have developed a general and completely automatic method (the method of *flag algebras*) that often succeeds for natural extremal problems; discussing it in more details is beyond the scope of this introduction.

We can of course ask the question we considered for triangles in more general setting: For a graph  $F$ , let  $\text{ex}(F; n)$  be the maximum possible number of edges of an  $n$ -vertex graph that does not contain  $F$  as a subgraph. For example, Theorem 48 states that  $\text{ex}(K_3; n) \leq n^2/4$ , and the observation with bipartite graphs gives  $\text{ex}(K_3; n) \geq \lfloor n^2/4 \rfloor$ . Since  $\text{ex}(K_3; n)$  is an integer, this shows that

$$\text{ex}(K_3; n) = \left\lfloor \frac{n^2}{4} \right\rfloor.$$

Analogously to Theorem 48, it is natural to guess that for a clique  $F = K_{k+1}$ , the maximum is given by (nearly) balanced complete  $k$ -partite graphs; and this is indeed the case.

**Theorem 49** (Turán). *For every  $k \geq 1$ , we have*

$$\text{ex}(K_{k+1}; n) \leq \left(1 - \frac{1}{k}\right) \cdot \frac{n^2}{2}.$$

Even more generally, this turns out to be the correct bound for all graphs  $F$  of chromatic number exactly  $k + 1$ . Here, the lower bound is clear:  $F$  cannot be a subgraph of a complete  $k$ -partite graph (since such a graph has a proper  $k$ -coloring), while proving the upper bound is substantially more complicated.

**Theorem 50** (Erdős-Stone). *For every graph  $F$  with  $\chi(F) = k + 1 \geq 2$ , we have*

$$\text{ex}(F; n) = \left(1 - \frac{1}{k}\right) \cdot \frac{n^2}{2} + o(n^2).$$

*That is, for every  $\varepsilon > 0$ , there exists  $n_0$  such that for every  $n \geq n_0$ , we have*

$$\text{ex}(F; n) \leq \left(1 - \frac{1}{k}\right) \cdot \frac{n^2}{2} + \varepsilon n^2.$$

Theorem 50 gives an asymptotically precise estimate of  $\text{ex}(F; n)$  when  $\chi(F) \geq 3$ . For a bipartite graph  $F$ , it only tells us that  $\text{ex}(F; n) = o(n^2)$ , but  $\text{ex}(F; n)$  can in fact be much smaller; we will see the most basic example of this next.

## 7.2 $C_4$ -free graphs

An upper bound on  $\text{ex}(C_4; n)$  can be obtained by a double-counting argument for the same quantity  $m(G)$  that we used in the proof of Theorem 48.

**Theorem 51.** *Every  $n$ -vertex graph  $G$  without 4-cycles has at most  $\frac{n^{3/2}}{\sqrt{2}}$  edges.*

*Proof.* Let  $G$  be any  $n$ -vertex  $C_4$ -free graph and let  $e$  be the number of edges of  $G$ . Recall that  $m(G)$  is the number of triples  $(u, v_1, v_2)$  of vertices of  $G$  such that  $uv_1, uv_2 \in E(G)$ . Observe that (7.1) and (7.4) from the proof of Theorem 48 do not assume anything about the graph, and thus they hold for  $G$  as well:

$$m(G) = \sum_{v \in V(G)} \deg^2 v = n \cdot \frac{\sum_{v \in V(G)} \deg^2 v}{n} \geq n \cdot \left(\frac{2e}{n}\right)^2 = \frac{4e^2}{n}.$$

On the other hand:

- There are at most  $2e$  triples  $(u, v_1, v_2)$  of vertices of  $G$  such that  $uv_1 \in E(G)$  and  $v_1 = v_2$ .

- For any distinct vertices  $v_1, v_2 \in V(G)$ , there is at most one vertex  $u$  such that  $uv_1, uv_2 \in E(G)$ , since if  $v_1$  and  $v_2$  had (at least) two distinct common neighbors  $u$  and  $u'$ , then  $G$  would contain the 4-cycle  $uv_1u'v_2$ . We can choose the vertices  $v_1 \neq v_2$  in exactly  $n(n-1)$  ways.

This implies that

$$m(G) \leq 2e + n(n-1) \leq 2\binom{n}{2} + n(n-1) \leq 2n^2.$$

By combining the two inequalities, we get

$$\frac{4e^2}{n} \leq m(G) \leq 2n^2,$$

and thus

$$e \leq \frac{n^{3/2}}{\sqrt{2}}.$$

□

The exponent  $3/2$  in this bound is correct, as shown by the following construction. Let  $\mathbb{F}$  be any finite field (say  $\mathbb{Z}_k$  for a prime  $k$ ), and let  $k = |\mathbb{F}|$ . Let  $G_{\mathbb{F}}$  be the bipartite graph with parts  $L$  and  $R$  such that

- $L$  is the set of all  $k^2$  functions of form

$$f(x) = ax + b$$

for constants  $a, b \in \mathbb{F}$ ,

- $R$  is the set of all  $k^2$  pairs  $(x, y) \in \mathbb{F}^2$ , and
- vertices  $f \in L$  and  $(x, y) \in R$  are adjacent if and only if  $y = f(x)$ .

Note that any two vertices  $f_1, f_2 \in L$  have at most one common neighbor, since the system

$$y = a_1x + b_1$$

$$y = a_2x + b_2$$

(where  $(a_1, b_1) \neq (a_2, b_2)$ ) has either exactly one solution (when  $a_1 \neq a_2$ ) or no solution (when  $a_1 = a_2$  and  $b_1 \neq b_2$ ). Therefore, the graph  $G_{\mathbb{F}}$  does not contain any 4-cycles. This graph clearly has  $2k^2$  vertices. Moreover, each vertex  $f \in L$  has exactly  $k$  neighbors  $\{(x, f(x)) : x \in \mathbb{F}\}$ , and thus  $G_{\mathbb{F}}$  has  $k^3$  edges.

**Corollary 52.** *For infinitely many integers  $n$ , we have*

$$\text{ex}(C_4; n) \geq (n/2)^{3/2}.$$

## 7.3 Homework

1. Let  $G$  be a graph with  $n$  vertices and  $m$  edges. For  $i \in \{1, 2, 3\}$ , let  $t_i$  be the number of subsets of  $V(G)$  of size three inducing a subgraph with exactly  $i$  edges (thus,  $t_3$  is the number of triangles in  $G$ ).

- By double-counting the number of pairs  $(e, v)$  such that  $e \in E(G)$ ,  $v \in V(G)$ , and the edge  $e$  is *not* incident with the vertex  $v$ , show that  $m(n - 2) = t_1 + 2t_2 + 3t_3$ .
- By double-counting the number of pairs  $(v, D)$  such that  $v$  is a vertex of  $G$  and  $D$  is a set consisting of two neighbors of  $v$ , show that

$$\sum_{v \in V(G)} \binom{\deg v}{2} = t_2 + 3t_3.$$

Use these identities to show that if  $G$  is triangle-free, then

$$m = \frac{n^2}{4} - \frac{t_1 + \sum_{v \in V(G)} (\deg v - n/2)^2}{n}.$$

2. Show that for all positive integers  $k$  and  $d$ ,

- every graph of minimum degree at least  $k - 1$  contains as subgraphs all trees with at most  $k$  vertices, and
- every  $n$ -vertex graph with more than  $(d - 1)n$  edges contains a subgraph of minimum degree at least  $d$ .

Use this to show that  $\text{ex}(T; n) \leq (|V(T)| - 2)n$  holds for every tree  $T$  with at least two vertices. What lower bound can you provide for  $\text{ex}(T; n)$ ?

3. Recall that  $\alpha(G)$  is the maximum size of an independent set in a graph  $G$  and that

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

is the average degree of  $G$ . Using Turán's theorem, show that

$$\alpha(G) \geq \frac{|V(G)|}{\bar{d}(G) + 1}$$

holds for every graph  $G$ .

## 7.4 Tutorial

1. Determine  $\text{ex}(K_2; n)$ ,  $\text{ex}(2K_2; n)$ , and  $\text{ex}(K_{1,2}; n)$ .
2. Find a graph  $G$  with the following property: Adding any possible edge to  $G$  creates a triangle, but  $|E(G)| < \text{ex}(K_3; |V(G)|)$ .
3. Show that
  - if  $F_1 \subseteq F_2$ , then  $\text{ex}(F_1; n) \leq \text{ex}(F_2; n)$  holds for every positive integer  $n$ , and that
  - for every graph  $F$  and for all integers  $n_2 \geq n_1 \geq 2$ ,

$$\frac{\text{ex}(F; n_1)}{\binom{n_1}{2}} \geq \frac{\text{ex}(F; n_2)}{\binom{n_2}{2}}.$$

Hint: Let  $G$  be a graph with  $n_2$  vertices and  $\text{ex}(F; n_2)$  edges such that  $F \not\subseteq G$ . Double-count the number of pairs  $(e, X)$ , where  $X$  is a subset of  $V(G)$  of size  $n_1$  and  $e$  is an edge of  $G[X]$ .

4. By double-counting the number of tuples  $(v, u_1, u_2, \dots, u_a)$  such that  $u_1, \dots, u_a$  are (not necessarily distinct) neighbors of  $v$ , show that  $\text{ex}(K_{a,b}; n) = O(n^{2-1/a})$  holds for all positive integers  $a$  and  $b$ . Use this to show that  $\text{ex}(F; n) = O(n^{2-1/\lfloor |V(F)|/2 \rfloor})$  holds for every bipartite graph  $F$ .
5. Let  $V$  be a set of size  $n$  and let  $k, r$ , and  $\lambda$  be non-negative integers. A system  $\mathcal{B} \subseteq \binom{V}{k}$  consisting of  $b$   $k$ -element subsets of  $V$  (called *blocks*) is a  $(n, b, r, k, \lambda)$ -*design* if
  - each element of  $V$  is contained in exactly  $r$  blocks, and
  - any pair of distinct elements of  $V$  is contained in exactly  $\lambda$  blocks.

Show that the system  $\binom{V}{k}$  of all  $k$ -element subsets of  $V$  is a  $(n, b, r, k, \lambda)$ -design for some parameters  $n, b, r, k$ , and  $\lambda$ , and determine the values of these parameters.

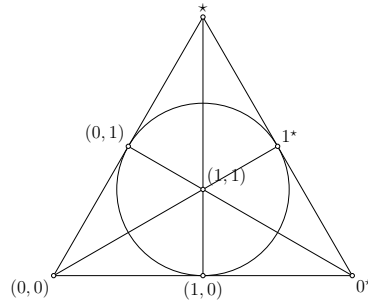
6. Let  $\mathbb{F}$  be a finite field of size  $q$ , let

$$V = \{(x, y) : x, y \in \mathbb{F}\} \cup \{x^* : x \in \mathbb{F}\} \cup \{\star\},$$

and let  $\mathcal{P}_{\mathbb{F}} \subseteq \binom{V}{q+1}$  be the system consisting of the following sets:

- $\{(x, ax + b) : x \in \mathbb{F}\} \cup \{a^*\}$  for all  $a, b \in \mathbb{F}$ ,
- $\{(a, y) : y \in \mathbb{F}\} \cup \{\star\}$  for all  $a \in \mathbb{F}$ , and
- $\{a^* : a \in \mathbb{F}\} \cup \{\star\}$ .

For example,  $\mathcal{P}_{\mathbb{Z}_2}$  consists of the sets of points joined by a straight line or a circle in the following picture:



Show that  $\mathcal{P}_{\mathbb{F}}$  is a  $(q^2 + q + 1, q^2 + q + 1, q + 1, q + 1, 1)$ -design. Remark:  $\mathcal{P}_{\mathbb{F}}$  is called the *finite projective plane over  $\mathbb{F}$* .

7. Let  $\mathcal{B} \subseteq \binom{V}{k}$  be a  $(n, b, r, k, \lambda)$ -design. By double-counting

- the pairs  $(v, B)$  such that  $v \in V$ ,  $B \in \mathcal{B}$ , and  $v \in B$ , and
- the pairs  $(D, B)$  such that  $D \in \binom{V}{2}$ ,  $B \in \mathcal{B}$ , and  $D \subseteq B$ ,

show that

$$b = \frac{rn}{k} = \frac{\lambda n(n-1)}{k(k-1)}.$$

## Lesson 8

# Introduction to Ramsey theory

### 8.1 Finite Ramsey theorem for pairs

Can you find a graph on five vertices that contains neither a clique nor an independent set of size three? And on six vertices? The latter corresponds to a well-known mathematical observation: On each party with at least six participants, there necessarily are either three people that have not met before, or three people that have met each other before.

More formally, for sets  $V$  and  $C$ , let  $\varphi : \binom{V}{2} \rightarrow C$  be a function assigning an element of  $C$  to each pair of vertices of  $V$ . Here are two possible graph-theoretic interpretations of such a function:

- If  $C = \{\text{edge}, \text{non-edge}\}$ , then  $\varphi$  corresponds to a graph on vertex set  $V$ , with two vertices  $u$  and  $v$  adjacent if and only if  $\varphi(\{u, v\}) = \text{edge}$ .
- We can view  $\varphi$  as describing a coloring of the edges of the complete graph on vertex set  $V$ : The color of the edge  $uv$  is  $\varphi(\{u, v\})$ .

We say that a set  $Z \subseteq V$  of size at least two is *monochromatic* or *homogeneous* if the restriction of  $\varphi$  to  $\binom{Z}{2}$  is a constant function, i.e., for all  $u, v, u', v' \in Z$  such that  $u \neq v$  and  $u' \neq v'$ , we have  $\varphi(\{u, v\}) = \varphi(\{u', v'\})$ . We can now state a famous theorem of Ramsey.

**Theorem 53** (Ramsey theorem). *For all integers  $c \geq 2$  and  $k \geq 1$ , there exists an integer  $n$  such that for every set  $V$  of size at least  $n$  and every  $c$ -coloring  $\varphi : \binom{V}{2} \rightarrow \{1, \dots, c\}$  of pairs of elements of  $V$ , there exists a monochromatic subset of  $V$  of size  $k$ . Moreover,  $n \leq c^{ck}$ .*

*Proof.* Let  $V$  be a set of size at least  $c^{ck}$  and let  $\varphi : \binom{V}{2} \rightarrow \{1, \dots, c\}$  be any  $c$ -coloring of pairs of elements of  $V$ . Let  $V_0 = V$ . Let us choose distinct elements  $v_1, \dots, v_{ck} \in V$ , colors  $m_1, \dots, m_{ck} \in \{1, \dots, c\}$  and sets  $V_{ck} \subset V_{ck-1} \subset \dots \subset$

$V_1 \subset V_0$  such that  $|V_i| \geq c^{ck-i}$  as follows: We proceed for  $i = 1, 2, \dots, ck$ , and thus when we consider given  $i$ , we already know  $V_{i-1}$ . Since  $|V_i| \geq c^{ck-(i-1)}$ , we have  $V_{i-1} \neq \emptyset$ ; we choose the element  $v_i \in V_{i-1}$  arbitrarily. Next, we look at the pairs  $\{v_i, x\}$  for  $x \in V_{i-1} \setminus \{v_i\}$ ; there necessarily exists a color  $m_i \in \{1, \dots, c\}$  such that at least  $c^{ck-i}$  of the pairs have color  $m_i$ : Indeed, otherwise there are at most  $c^{ck-i} - 1$  such pairs of each color, and thus  $|V_{i-1}| \leq 1 + c \cdot (c^{ck-i} - 1) = c^{ck-(i-1)} + 1 - c < c^{ck-(i-1)}$ , which is a contradiction. We let  $V_i = \{x \in V_{i-1} \setminus \{v_i\} : \varphi(\{v_i, x\}) = m_i\}$ .

Note that for every  $i, j \in \{1, \dots, ck\}$  such that  $i < j$ , we have  $\varphi(\{v_i, v_j\}) = m_i$ ; indeed, the construction ensures that  $\varphi(\{v_i, x\}) = m_i$  for every  $x \in V_i$ , and we have  $v_j \in V_{j-1} \subseteq V_i$ . Obviously, there exists a set  $I \subseteq \{1, \dots, ck\}$  of size  $k$  and a color  $m \in \{1, \dots, c\}$  such that  $m_i = m$  for every  $i \in I$ . But then the set  $\{v_i : i \in I\}$  is monochromatic, since all pairs of elements from this set have color  $m$ .  $\square$

Intuitively, this says that “perfect chaos is impossible”: In any coloring of edges of a huge complete graph by a bounded number of colors, there will be a large perfectly orderly (homogeneous) part. How large  $n$  we actually need in Ramsey theorem? Let us establish a commonly used notation:

$$n \rightarrow (k)_c^2$$

is a shortcut for the statement “for every  $c$ -coloring of pairs of elements of a set of size  $n$ , there exists a monochromatic subset of size  $k$ ”. The  $c$ -color Ramsey number  $R_c(k)$  of an integer  $k$  is then defined as the minimum integer  $n$  such that  $n \rightarrow (k)_c^2$ . The term *Ramsey number*  $R(k)$  refers to the case  $c = 2$ .

Theorem 53 implies that  $R(k) \leq 2^{2k}$ . How far is this from the right value? Erdős came up with the following probabilistic lower bound.

**Lemma 54.** *Let  $k \geq 2$  be an integer and let  $V$  be a set of size  $n \leq 2^{(k-1)/2}$ . Let  $G$  be the random graph with vertex set  $V$ , obtained by joining each pair of vertices of  $V$  by an edge independently at random with probability 50%. Then the probability that  $\alpha(G) < k$  and  $\omega(G) < k$  is non-zero.*

*Proof.* Let  $h$  be the number of homogeneous subsets of  $V$  in  $G$  (cliques or independent sets) of size exactly  $k$ . By the linearity of expectation, we have

$$\mathbb{E}[h] = \sum_{X \in \binom{V}{k}} \Pr[X \text{ is a homogeneous set in } G].$$

Any  $k$ -element set  $X$  is homogeneous if and only if either all pairs of vertices from  $X$  are adjacent, or they are all non-adjacent. Since each pair of vertices is independently adjacent with probability 50%, the probability that all  $\binom{k}{2}$  pairs are adjacent is  $2^{-\binom{k}{2}}$ , and the probability that they are all non-adjacent is the same. Hence,

$$\Pr[X \text{ is a homogeneous set in } G] = 2 \cdot 2^{-\binom{k}{2}}.$$

It follows that

$$\mathbb{E}[h] = \binom{n}{k} \cdot 2 \cdot 2^{-\binom{k}{2}} < n^k \cdot 2^{-\binom{k}{2}} = \left(n \cdot 2^{(k-1)/2}\right)^k \leq 1.$$

Since  $\mathbb{E}[h] < 1$  and  $h$  is an integer, we must have  $h = 0$  with non-zero probability. However,  $h = 0$  is equivalent to the claim that  $\alpha(G) < k$  and  $\omega(G) < k$ .  $\square$

Thus, if  $n \leq 2^{(k-1)/2}$ , then  $n \not\rightarrow (k)_2^2$ ; and thus  $R(k) > 2^{(k-1)/2}$ . In conclusion, we know that

$$2^{(k-1)/2} < R(k) \leq 2^{2k} = 4^k$$

holds for every integer  $k \geq 2$ . These bounds are rather far apart (note that  $2^{2k} = (2^{k/2})^4$ ). We do not know any substantially better lower bounds, and the upper bound has only very recently been substantially improved (to roughly  $3.78^k$ ).

Let us also remark that the proof of Lemma 54 can be modified to show that almost all of the graphs on roughly  $2^{k/2}$  vertices have both independence number and clique number less than  $k$ ; however, it is quite difficult to find an explicit construction of such a graph.

## 8.2 Variations on the Ramsey theorem

Ramsey theorem inspired many other analogous results; let us go over a few of these variations. Instead of coloring pairs of elements, we can color  $s$ -tuples for any integer  $s \geq 3$ . The definitions extend naturally to this setting: For a set  $V$ , a  $c$ -coloring of  $s$ -tuples from  $V$  is any function  $\varphi : \binom{V}{s} \rightarrow \{1, \dots, c\}$ , and a set  $X \subseteq V$  is *monochromatic* if  $\varphi$  assigns the same color to all  $s$ -tuples of elements of  $V$ . We write

$$n \rightarrow (k)_c^s$$

is a shortcut for the statement “for every  $c$ -coloring of  $s$ -tuples of elements of a set of size  $n$ , there exists a monochromatic subset of size  $k$ ”.

**Theorem 55** (Ramsey theorem for  $s$ -tuples). *For all integers  $s, c \geq 2$  and  $k \geq 1$ , there exists an integer  $n$  such that  $n \rightarrow (k)_c^s$ .*

This can be proved similarly to Theorem 53 by induction on  $s$ ; we leave this for the tutorials. This version of the theorem is rather useful, since many other variations can be proved by “encoding” them in a coloring of sufficiently large tuples. For instance, consider the following geometric Ramsey statement. A set of points in the plane is in *generic position* if no three of the points lie on a straight line.

**Lemma 56.** *For every integer  $k \geq 3$ , there exists an integer  $n$  such that any set of  $n$  points in the plane in generic position contains  $k$  points in convex position.*

*Proof.* Choose  $n$  so that  $n \rightarrow (k)_2^3$ ; such an integer  $n$  exists by Theorem 55. Let  $V$  be any set of  $n$  points in the plane in generic position; without loss of generality, we can assume that no two of them have the same  $x$ -coordinate (otherwise, we can rotate the plane). Let  $v_1, \dots, v_n$  be the points of  $V$  sorted by the  $x$ -coordinate. Let  $\varphi : \binom{V}{3} \rightarrow \{\text{up}, \text{down}\}$  be defined as follows: For all  $i < j < k$ , if  $v_j$  is above the straight line joining the points  $v_i$  and  $v_k$ , then  $\varphi(\{v_i, v_j, v_k\}) = \text{up}$ , otherwise  $\varphi(\{v_i, v_j, v_k\}) = \text{down}$ . Since  $n \rightarrow (k)_2^3$ , there exists a subset  $X \subseteq V$  of size  $k$  monochromatic in  $\varphi$ . Observe that the points of  $X$  must be in convex position.  $\square$

Another common option is to consider infinite versions of Ramsey theorem. The very basic one is as follows.

**Theorem 57** (Infinite Ramsey theorem for  $s$ -tuples). *For all integers  $s, c \geq 2$ , for any  $c$ -coloring of the  $s$ -tuples of elements of any infinite set  $V$ , there exists an infinite monochromatic set  $X \subseteq V$ .*

This can be proved analogously to Theorem 53, with the modification that we construct an infinite sequence  $v_1, v_2, \dots$  and an infinite chain  $V \supset V_1 \supset V_2 \supset \dots$  of infinite subsets.

### 8.3 Homework

1. Show that for every positive integer  $c$ , there exists  $n$  such that the following claim holds. For any coloring  $\varphi : \{1, \dots, n\} \rightarrow \{1, \dots, c\}$  of the first  $n$  integers using  $c$  colors, there exist numbers  $x, y \in \{1, \dots, n\}$  such that  $x + y \leq n$  and  $\varphi(x) = \varphi(y) = \varphi(x + y)$ . Hint: Consider the complete graph with vertex set  $\{1, \dots, n\}$  and color each edge  $uv$  (where  $u < v$ ) by the color  $\varphi(v - u)$ . Look at a monochromatic triangle in this graph.
2. Show that for every integer  $k \geq 3$ , there exists an integer  $n$  such that the following claim holds: For any coloring  $\varphi$  of edges of the complete graph  $G$  with vertex set  $\{1, \dots, n\}$  (using any number of colors), there exists a set  $X \subseteq V(G)$  of size  $k$  such that either
  - $\varphi$  assigns the same color to all edges between vertices of  $X$ , or
  - there are no vertices  $x, y, z \in X$  such that  $x < y < z$  and  $\varphi(xy) = \varphi(yz)$ .

Hint: Give each triple  $\{u, v, w\}$  such that  $u < v < w$  a color based on whether  $\varphi(uv) = \varphi(vw)$  and apply the Ramsey theorem for triples to obtain a monochromatic set of size  $k + 1$ .

3. Prove that for every positive integer  $c$  and every  $c$ -coloring  $\varphi : \binom{V}{2} \rightarrow \{1, \dots, c\}$  of 2-tuples of elements of an infinite set  $V$ , there exists an infinite monochromatic subset of  $V$ . I.e. prove the case  $s = 2$  of Theorem 57 from the lecture notes. Hint: Modify the proof of Theorem 53 from the lecture notes.

## 8.4 Tutorial

1. Let  $R(\alpha, \omega)$  be the smallest integer  $n$  such that every graph with at least  $n$  vertices contains an independent set of size  $\alpha$  or a clique of size  $\omega$ . Determine the values  $R(\alpha, 1)$ ,  $R(\alpha, 2)$ ,  $R(1, \omega)$ , and  $R(2, \omega)$ , and show that for  $\alpha, \omega \geq 2$ , we have

$$R(\alpha, \omega) \leq R(\alpha - 1, \omega) + R(\alpha, \omega - 1).$$

2. Use this to show that

$$R(\alpha, \omega) \leq \binom{\alpha + \omega - 2}{\alpha - 1}$$

holds for all positive integers  $\alpha$  and  $\omega$ . What bound does this give you for  $R_2(k)$ ?

3. Generalize the idea from the previous two exercises to give a bound on  $R_c(k)$  for any number  $c \geq 3$  of colors.
4. Let  $\varphi : \binom{V}{3} \rightarrow \{1, \dots, c\}$  be a  $c$ -coloring of 3-tuples of vertices from some set  $V$ . Let us run the following procedure as long as there are any vertices left: We choose a vertex  $v$  arbitrarily, and let  $X \subseteq V \setminus \{v\}$  be the largest set such that  $\varphi$  assigns the same color to all triples  $\{x, y, v\}$  for distinct  $x, y \in X$ . Delete all the vertices not in  $X$ , and repeat. Show that for every positive integer  $m$ , if the initial size of  $V$  is at least  $1 + R_c(1 + R_c(1 + R_c(\dots(1)\dots)))$  (nested  $m$  times), then this procedure will run for at least  $m$  rounds.
5. Use this to prove that for all integers  $c, k \geq 2$ , there exists an integer  $n$  such that  $n \rightarrow (k)_c^3$  (i.e., that Theorem 55 from the lecture notes holds for  $s = 3$ ).
6. Show that for every positive integer  $k$ , there exists  $n$  such that any 2-coloring  $\varphi$  of the edges of  $K_{n,n}$  contains a monochromatic subgraph isomorphic to  $K_{k,k}$ . Hint: Let  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$  be the vertices forming the two parts of  $K_{n,n}$ . Consider the complete graph with vertex set  $v_1, \dots, v_n$  and with each edge  $v_i v_j$  (where  $i < j$ ) receiving the color  $(\varphi(x_i y_j), \varphi(x_j y_i))$ .
7. Let  $G$  be the infinite complete bipartite graph with parts formed by vertices  $x_1, x_2, \dots$  and  $y_1, y_2, \dots$  and let us color each edge  $x_i y_j$  blue if  $i \leq j$  and red otherwise. Show that  $G$  contains a monochromatic subgraph isomorphic to  $K_{k,k}$  for arbitrarily large finite  $k$ , but there are no infinite sets  $X \subseteq \{x_1, x_2, \dots\}$  and  $Y \subseteq \{y_1, y_2, \dots\}$  such that all edges  $xy$  with  $x \in X$  and  $y \in Y$  have the same color.

## Lesson 9

# Introduction to random graph theory

Why might we be interested in the behavior of random graphs? There are a number of reasons.

- Sometimes, it is hard to find an explicit example of a graph with a certain property, but easy to show that a random graph satisfies it (it is difficult to “find hay in a haystack”). We have seen an example in the previous lecture when we gave a lower bound on the Ramsey numbers.
- In software testing, we often do not have real data available, or we have it available only in a limited quantity. A natural idea is then to generate additional testing data at random. For that to be useful, we however need to know that such randomly generated data have reasonably similar properties to the real data to which the software will be applied.
- A similar remark applies to other scenarios where we need to evaluate something (e.g., the efficacy of various interventions during the covid era) without being able to run experiments on the real system.
- Data analysis: Given a number of properties observed in a system, which are the “fundamental” ones and which are just their consequences? One way to approach this question is to create a random model so that it satisfies some of the properties and check whether it also satisfies the rest of them.

### 9.1 Erdős-Rényi graphs

There are of course many ways how to construct a graph at random. In the previous lecture, we have seen the most basic one: For each pair of vertices, we independently flip a fair coin and determine whether to join them by an

edge depending on an outcome. Since each pair of vertices is adjacent with probability  $1/2$ , this results in a graph where the expected degree of any fixed vertex is  $\frac{n-1}{2}$ , where  $n$  is the number of vertices.

When we want to consider sparser graphs, instead of using a fair coin, we can use a biased one: Let  $p \in [0, 1]$  be a real number and  $n$  an integer. We fix a set  $V$  of  $n$  vertices and for every pair of vertices, we independently at random join them by an edge with probability  $p$ . This model is called *Erdős-Rényi random graph with density  $p$*  and the resulting random graph is denoted by  $G(n, p)$ .

Formally, a *random graph* is rather a probability distribution on graphs. For example,  $G(n, p)$  is the probability distribution on graphs with vertex set  $V$  assigning to each such graph  $H$  the probability that it arises from the described process; in this case,

$$\Pr[G(n, p) = H] = p^{|E(H)|} (1 - p)^{\binom{n}{2} - |E(H)|}.$$

The statements such as “the expected number of triangles in  $G(n, p)$  is  $p^3 \binom{n}{3}$ ” should be interpreted as a shorthand for the claim that the expectation of the number of triangles in a graph chosen from this probability distribution is  $p^3 \binom{n}{3}$ .

Let us remark that in the case  $p = 1/2$ , i.e., of the Erdős-Rényi random graph  $G(n, 1/2)$ , the probability that a fixed graph  $H$  is chosen is  $2^{-\binom{n}{2}}$ , that is, the same for every  $n$ -vertex graph  $H$ . In other words,  $G(n, 1/2)$  is the uniform distribution on  $n$ -vertex graphs, and showing that  $G(n, p)$  has some property with high probability is the same as saying that most of the  $n$ -vertex graphs have this property.

One interesting phenomena that we observe in (Erdős-Rényi) random graphs are the *phase transitions*: For many properties, there exists a threshold  $p_0$  such that  $G(n, p)$  has the property almost surely when  $p > p_0$  and it is very unlikely to have this property when  $p < p_0$ . As perhaps the most fundamental example, let us consider the components of  $G(n, p)$ .

**Theorem** (Giant Component Theorem). *Let  $p : \mathbb{N} \rightarrow [0, 1]$  be any function. For every  $\varepsilon > 0$ ,*

- *if  $p(n) < \frac{1-\varepsilon}{n}$  for every  $n$ , then*

$$\lim_{n \rightarrow \infty} \Pr [\text{the largest component of } G(n, p(n)) \text{ has size } O(\log(n))] = 1$$

- *if  $p(n) = 1/n$ , then*

$$\lim_{n \rightarrow \infty} \Pr \left[ \text{the largest component of } G(n, p(n)) \text{ has size } \Theta(n^{2/3}) \right] = 1$$

- *if  $p(n) > \frac{1+\varepsilon}{n}$  for every  $n$ , then*

$$\lim_{n \rightarrow \infty} \Pr \left[ \begin{array}{l} \text{the largest component of } G(n, p(n)) \text{ has size} \\ \Omega(n), \text{ all other components } O(\log(n)) \end{array} \right] = 1$$

- *if  $p(n) < \frac{(1-\varepsilon)\log n}{n}$  for every  $n$ , then*

$$\lim_{n \rightarrow \infty} \Pr [G(n, p(n)) \text{ contains at least one isolated vertex}] = 1$$

- *if  $p(n) > \frac{(1+\varepsilon)\log n}{n}$  for every  $n$ , then*

$$\lim_{n \rightarrow \infty} \Pr [G(n, p(n)) \text{ is connected}] = 1$$

We do not really have the tools (or time) to prove this, but let us at least get an intuition beyond the first of the transitions. Suppose that  $p(n) = c/n$  and imagine that we want to find the component containing one particular vertex  $v_0$  of  $G(n, p)$ . We can do so by “revealing  $G(n, p)$  step by step”: We go through all pairs  $\{v_0, u\}$  and look at the corresponding coin flips to determine which of them are edges, thus determining the neighbors of  $v_0$ . Then we pick one of these neighbors (call it  $v_1$ ), go through all pairs  $\{v_1, u\}$  (where  $u \neq v_0$ ) and see which of them are edges, revealing additional vertices of the component. We proceed like this (taking the not-yet-processed vertices of the components and revealing their neighbors) until we have seen the whole component. Suppose we have already revealed  $k$  of the vertices of the component, where  $k$  is still much smaller than  $n$ , and we are revealing neighbors of a vertex  $v$ . In this case, we are considering  $n - k \approx n$  pairs of vertices and each of them has probability  $c/n$  to be an edge, and thus the expected number of revealed neighbors is almost exactly  $c$ . That is, we have processed one vertex and added in expectation  $c$  new not-yet-processed ones, and consequently the number of not-yet-processed vertices changed by  $c - 1$  in expectation. That means that if  $c \leq 1 - \varepsilon$ , then number of not-yet-processed vertices will be in expectation decreasing with each step by  $\varepsilon$ , and thus we reach zero not-yet-processed vertices (and finish the component) in roughly  $1/\varepsilon$  steps. Of course, some components may be “lucky”

and beat this bound; but a more careful analysis shows that this probability decreases exponentially with the number of steps, and the probability that the component has much more than  $\log n$  vertices turns out to be  $o(1/n)$ . Therefore, since there are at most  $n$  components in total, the probability that any of them has substantially more than  $\log n$  vertices goes to 0 as  $n$  increases.

For constant  $p$ , all vertices of  $G(n, p)$  have (with probability going to 1 as  $n \rightarrow \infty$ ) degree  $pn + O(\sqrt{n \log n})$ ; let us prove a slightly weaker result for the case  $p = 1/2$ .

**Lemma 58.** *For every  $\varepsilon > 0$  there exists  $c > 1$  such that the probability that  $G(n + 1, 1/2)$  contains a vertex of degree less than  $n/2 - \varepsilon n$  (or more than  $n/2 + \varepsilon n$ ) is at most*

$$\frac{n + 1}{c^n}.$$

*Proof.* By symmetry, it suffices to consider the case of vertex of degree less than  $n/2 - \varepsilon n$ . Let  $v$  be a fixed vertex of  $G(n + 1, 1/2)$ ; we have

$$\Pr[\deg v < n/2 - \varepsilon n] = \frac{\sum_{0 \leq d < n/2 - \varepsilon n} \binom{n}{d}}{2^n} \leq \frac{2^{nH(\frac{n/2 - \varepsilon n}{n})}}{2^n} = 2^{n(H(1/2 - \varepsilon) - 1)},$$

i.e., at most  $c^{-n}$  for the constant  $c = 2^{1 - H(1/2 - \varepsilon)} > 1$ . Since this holds for all  $n + 1$  vertices of the graph, the probability that any of them has degree less than  $n/2 - \varepsilon n$  is at most  $\frac{n+1}{c^n}$ .  $\square$

Similarly, we can argue that the total number of edges of  $G(n, p)$  is almost surely close to  $p\binom{n}{2}$ . However, sometimes it may be convenient to work with random graphs with a precisely known number of edges, that is, with the following important variation on Erdős-Rényi graphs: Let  $G'(n, m)$  denote the uniformly random  $n$ -vertex graph with exactly  $m$  edges (intuitively, this random graph should be quite close to  $G(n, \frac{m}{\binom{n}{2}})$  in terms of its properties, but of course different). More precisely, for a fixed set  $V$  of  $n$  vertices,  $G'(n, m)$  is the probability distribution on graphs with vertex set  $V$  such that for any fixed graph  $H$  on this vertex set,

$$\Pr[G'(n, m) = H] = \frac{1}{\binom{\binom{n}{2}}{m}}$$

if  $|E(H)| = m$  and  $\Pr[G'(n, m) = H] = 0$  otherwise. We will leave the question of how to generate such a graph efficiently for the tutorials.

## 9.2 Configuration model

Can we get a random graph where not only the number of edges is fixed, but actually all vertices have the same degree  $d$ ? Let us have a look at this question for small  $d$ . For  $d = 1$ , we simply want to generate a random *matching* between the  $n$  vertices, and this is not hard. E.g., we can simply order the vertices at

random, then join by edges the first one with the second one, the third one with the fourth one,  $\dots$ , and the next to last one with the last one. Let us remark that this construction implies that there are exactly

$$\mu(n) = \frac{n!}{(n/2)!2^{n/2}}$$

matchings on  $n$  vertices—there are  $n!$  ways to order the vertices, and the same matching will be generated for the orderings which differ only by reordering the matched pairs (in  $(n/2)!$  possible ways) and switching the orders within pairs (in  $2^{n/2}$  possible ways). Of course, in all of this, we need to assume that  $n$  is even.

For larger  $d$ , this is not so straightforward, but we can use the following trick, resulting in so called *configuration model*: Divide each of the  $n$  vertices into  $d$  parts, and pick a random matching  $M$  on the resulting  $dn$  vertices. Then contract each  $d$ -tuple of vertices back into the original vertex. It is easy to check that this produces every  $d$ -regular graph on these  $n$  vertices with the same probability. However, there is an issue that we need to deal with: It could happen that the resulting graph has parallel edges, in case  $M$  contains more than one edges between two of the  $d$ -tuples. It could even have loops, in case  $M$  joins two of the vertices within the same  $d$ -tuple. We do expect these things to happen, but not too often. For example, let us consider the loops.

**Lemma 59.** *The expected number of loops in a random  $d$ -regular graph on  $n$  vertices generated using the configuration model is*

$$\frac{d(d-1)n}{2(dn-1)} \approx \frac{d-1}{2}.$$

A more detailed analysis shows that the probability that the resulting graph contains neither loops nor parallel edges goes to  $e^{-(d^2-1)/4}$  as  $n$  increases, i.e., it is constant for any fixed  $d$ . Thus, if we want to generate a simple random  $d$ -regular graph, we just need to repeat the construction (on average  $e^{(d^2-1)/4}$  times) until we hit a simple graph. Let us remark that a similar procedure can be used more generally to generate random graphs where the vertices have prescribed degrees (not necessarily all the same).

### 9.3 More realistic models

The random graphs we have discussed so far usually are not very useful as models for large real-world networks, such as

- the web graph, whose vertices are the webpages and the edges correspond to hyperlinks between them,
- the facebook graph, whose vertices are facebook users and the edges correspond to the friendships,

- the real-world acquaintance graph,
- the network of neurons in the brain,
- the road network in a country, etc.

These graphs are relatively sparse (the number of edges is linear or close to linear in the number  $n$  of vertices), but the distances between vertices are on average small, say at most logarithmic in  $n$  (famously, any two people chosen at random are likely to be able to reach one another through a chain of around six acquaintances). These properties are satisfied by both Erdős-Rényi graphs (with adjacency probability linear in  $1/n$ ) and the regular graphs arising from the configuration model. However, the next property is harder to achieve.

The degrees of vertices in many of the real-world networks follow so-called *power law*, that is, the fraction of vertices of degree  $d$  is roughly proportional to  $d^{-\gamma}$  for a fixed constant  $\gamma$  (whose value typically is between 2 and 3). This is indicative of a “rich get richer” type of phenomena (e.g., a web page which is linked to from many other places is easier to find, and so it will gain additional links faster than an equivalent page which was initially linked to from fewer places). This is very different from sparse Erdős-Rényi graphs  $G(n, c/n)$ , where the degrees follows the Poisson distribution, i.e., the fraction of vertices of degree  $d$  is proportional to  $c^d/d!$  and the proportion of large-degree vertices is substantially smaller.

One of random models that addresses this issue is the *Barabási-Albert model*: We start from a fixed  $m$ -vertex graph and add the rest of the vertices one by one. When a vertex  $v$  is added, we choose  $m$  of the already existing vertices at random, where the probability that a vertex is chosen is proportional to its current degree (i.e., larger-degree vertices are more likely to be chosen), and add edges between these vertices and  $v$ . This results in a graph with approximately  $mn$  edges (i.e., average degree close to  $2m$ ) and where the fraction of degree  $d$  vertices is proportional to  $d^{-3}$ , i.e., follows power law.

Another property the real-world networks also exhibit is *clustering*: Two neighbors of a vertex are more likely to be adjacent than two random vertices (e.g., two of your friends are much more likely to know each other than two random people). In Erdős-Rényi model, the neighbors are no more likely to be adjacent than any two vertices, and this is essentially also true in the configuration model and Barabási-Albert model. As you could expect, researchers have developed random graph models that exhibit this property as well, but their discussion is beyond our scope.

## 9.4 Homework

1. For a graph  $G$ , let  $c_4(G)$  be the number of ordered 4-tuples of distinct vertices  $(v_1, v_2, v_3, v_4)$  of  $G$  such that  $v_1v_2, v_2v_3, v_3v_4, v_1v_4 \in E(G)$ . That is,  $c_4(G)$  counts the number of 4-cycles in  $G$  with a fixed starting vertex and a fixed orientation.

Let  $n, m \geq 4$  be integers such that  $m \leq \binom{n}{2}$  and let  $p = \frac{m}{\binom{n}{2}}$ . Compute and compare the expected values of  $c_4(G(n, p))$  and  $c_4(G'(n, m))$ .

2. Let  $p \in (0, 1)$  be a real number. Show that for every  $\varepsilon > 0$  there exists  $c > 1$  such that the probability that  $G(n+1, p)$  contains a vertex of degree less than  $(p-\varepsilon)n$  is at most  $\frac{n+1}{c^n}$ . To solve this exercise, use the inequality given at the end of the tutorial's assignment.
3. Consider the following random process: We start with  $2n$  isolated vertices labelled  $1, 2, \dots, 2n$ . For  $i = 1, 2, \dots, n$ , we choose a pair  $\{u_i, v_i\}$  uniformly at random among the  $\binom{2(n-(i-1))}{2}$  pairs of vertices belonging to  $\{1, 2, \dots, 2n\} \setminus \{u_1, v_1, \dots, u_{i-1}, v_{i-1}\}$ . Let  $M$  be the matching consisting of the edges  $\{u_1, v_1\}, \dots, \{u_n, v_n\}$ . Show that  $M$  is a uniformly random matching; that is, that for every fixed matching  $M'$  of  $\{1, \dots, 2n\}$ , the probability that  $M = M'$  is the same.

## 9.5 Tutorial

1. Determine the probability distributions corresponding to the random graphs  $G(3, 2/3)$  and  $G'(3, 2)$ ; that is, for each graph  $H$  with vertex set  $\{1, 2, 3\}$ , determine the probability that  $G(3, 2/3) = H$  and that  $G'(3, 2) = H$ .
2. Let  $n \geq 1$  and  $m \geq 0$  be integers such that  $m \leq \binom{n}{2}$ . Consider the following random process: We start with  $n$  isolated vertices labelled  $1, 2, \dots, n$ . For  $i = 1, 2, \dots, m$ , we choose a pair  $\{u_i, v_i\}$  uniformly at random among the  $\binom{n}{2} - (i-1)$  pairs of vertices different from  $\{u_1, v_1\}, \dots, \{u_{i-1}, v_{i-1}\}$ . Show that the random graph with vertex set  $\{1, \dots, n\}$  and with edge set  $\{\{u_1, v_1\}, \dots, \{u_m, v_m\}\}$  is equal to  $G'(n, m)$ ; that is, for every graph  $H$  with vertex set  $\{1, \dots, n\}$ , the probability that  $G'(n, m) = H$  is the same as the probability that the graph arising from this process is equal to  $H$ .
3. Let  $n \geq 4$  and  $m \geq 6$  be integers such that  $m \leq \binom{n}{2}$ , and let  $p = \frac{m}{\binom{n}{2}}$ . Compute and compare the expected number of cliques of size 4 in  $G(n, p)$  and in  $G'(n, m)$ . Hint: Make use of the equality

$$\frac{\binom{M-k}{m-k}}{\binom{M}{m}} = \prod_{i=0}^{k-1} \frac{m-i}{M-i} = \left(\frac{m}{M}\right)^k \prod_{i=0}^{k-1} \left(1 - \frac{i(M-m)}{m(M-i)}\right).$$

4. Consider the following random process. We start with the value 0. In each step, we increase the value by 1 with probability  $1/3$  and decrease it by 1 with probability  $2/3$ . The process stops once the value becomes negative. Show that the probability that this process runs for at least  $2s$  steps is at most

$$2^{-2H(1/2||2/3)s} = 1.125^{-s}.$$

Use this to show that if we run this process  $n$  times, the probability that it always runs for at most  $O(\log n)$  steps goes to 1 as  $n$  increases. How does this relate to the content of the lecture?

5. Let  $c > 0$  be a real number and let  $d$  be a non-negative integer. Show that the probability that a fixed vertex of  $G(n+1, c/n)$  has degree exactly  $d$  goes to

$$\frac{c^d e^{-c}}{d!}$$

as  $n$  increases.

For the last two exercises, you can use the following facts. For real numbers  $p, q \in [0, 1]$ , we define

$$H(q||p) = q \log_2 \frac{q}{p} + (1-q) \log_2 \frac{1-q}{1-p}.$$

As a remark, you can compare this to the definition of the entropy function. More importantly, note that  $H(q||p) \geq 0$ , with equality if and only if  $p = q$ . For all integers  $k \geq 0$  and  $n \geq k$  such that  $\frac{k}{n} \leq p$ , the following inequality holds:

$$\frac{1}{n+1} 2^{-H(k/n||p) \cdot n} \leq \binom{n}{k} p^k (1-p)^{n-k} \leq \sum_{i=0}^k \binom{n}{i} p^i (1-p)^{n-i} \leq 2^{-H(k/n||p) \cdot n}.$$

If you are interested, as an exercise for what we learned in Lesson 3, you can try to prove this inequality. Moreover, for every integer  $d \leq n$  and every  $p \in [0, 1]$ , we have

$$\frac{(np)^d}{d!} \cdot (1 - d^2/n) \cdot (1-p)^n \leq \binom{n}{d} p^d (1-p)^{n-d} \leq \frac{(np)^d e^{-p(n-d)}}{d!}.$$

## Part IV

# Connectivity in graphs

## Lesson 10

# Vertex and edge connectivity

Given a network (of railroads, computers, ...), it is natural to consider how robust it is; if part of the network (a few of the railroads or computers) fails, will the network still be usable? We may view this question from the perspective of the user (how often will I have to solve some emergency situation) or from the perspective of an attacker (what is the least amount of damage I need to do to make things break down).

Of course, as is often the case, there are many different interpretations of this question. Are we considering random failures, or ones caused by an intelligent attacker? What exactly does it mean for the network to still be usable – is it enough for majority of the nodes to be connected, or should everyone stay connected, or even should everyone still be connected by a reasonably short path? In this lecture, we are going to consider perhaps the simplest interpretation: Given a (connected) graph, what is the smallest number of edges (or vertices) we need to remove in order to break the graph into more than one component?

The result we will be aiming for in this setting is that if a graph is hard to disconnect, then there exist many disjoint paths between any pair of its vertices. In other words, not only are we guaranteed that the vertices will be connected by some path after any small number of failures, we can keep such a system of specific paths between them, and since the paths are disjoint, at least one of them will remain usable!

### 10.1 Edge connectivity

Let us start with the question concerning edge removal, which is somewhat more straightforward than the vertex removal version. For a non-negative integer  $k$ , we say that a graph  $G$  with at least two vertices is  *$k$ -edge-connected* if  $G - X$  is connected for every set  $X \subseteq E(G)$  of size less than  $k$ . Thus,

- every graph is 0-connected,
- a graph is 1-edge-connected if and only if it is connected, and
- a graph is 2-edge-connected if it stays connected even after the removal of any one of its edges.

The *edge-connectivity*  $\lambda(G)$  of  $G$  is the largest non-negative integer  $k$  such that  $G$  is  $k$ -edge-connected. Note that  $\lambda(G) \leq |E(G)|$ , because of the assumption that  $G$  has at least two vertices.

A set  $X \subseteq E(G)$  is an *edge-cut* if  $G - X$  is not connected. Thus, the edge-connectivity of  $G$  can also be defined as the minimum size of an edge-cut in  $G$ . A *k-edge-cut* is an edge-cut of size exactly  $k$ . We say that the edge-cut  $X$  *separates* vertices  $u$  and  $v$  of  $G$  if  $u$  and  $v$  belong to different components of  $G - X$ .

Clearly the edges incident with the same vertex always form an edge-cut, which gives us the following claim.

**Observation 60.** *Every graph  $G$  with at least two vertices satisfies  $\lambda(G) \leq \delta(G)$ .*

We have not defined these notions for the graph with only one vertex, which is impossible to disconnect by deletion of edges. It will be convenient for us to define that this graph is  $k$ -edge-connected for every positive integer  $k$ , but that its edge-connectivity is 0 (but other choices are definitely possible and appear in the literature).

Note that you have already seen a similar notion in the algorithmic context, of flows in networks, with the main difference that there one considered directed graphs with capacities on edges. In this context, the important result is that the size of the largest flow is equal to the minimum capacity of a cut. We are only going to use the version where all edges have capacity 1. Thus, for us, a *network* is a triple  $(\vec{G}, s, t)$ , where  $\vec{G}$  is a directed graph and  $s$  and  $t$  are distinct vertices of  $\vec{G}$ . For a function  $f : E(\vec{G}) \rightarrow \mathbb{R}$  and a vertex  $v \in V(\vec{G}) \setminus \{s, t\}$ , we define the *f-balance* of  $v$  be

$$\partial_f v = \sum_{e=(v,\star) \in E(\vec{G})} f(e) - \sum_{e=(\star,v) \in E(\vec{G})} f(e)$$

to be the difference of the amount given by  $f$  to the incoming and to the outgoing edges at  $v$ . We say that  $f$  is a *flow* if

- every edge  $e \in E(\vec{G})$  satisfies  $0 \leq f(e) \leq 1$  and
- every vertex  $v \in V(\vec{G}) \setminus \{s, t\}$  satisfies the *flow conservation condition*  $\partial_f v = 0$ ; that is, nothing is created or lost in  $v$ .

The *size* of the flow is  $\partial_f s$  (which is equal to  $-\partial_f t$ ). We also say that  $f$  is an *integral flow* if  $f(e) \in \{0, 1\}$  holds for every edge  $e$  of the network. A *cut* in the network is a set  $S \subset V(\vec{G})$  such that  $s \in S$  and  $t \notin S$ , and the *capacity* of this

cut is the number of edges of  $\vec{G}$  starting in  $S$  and ending in  $V(\vec{G}) \setminus S$ . We need the following fundamental result relating flows and cuts.

**Theorem 61** (Min-cut max-flow theorem). *For any network  $(\vec{G}, s, t)$ , the maximum size of a flow in the network is equal to the minimum capacity of a cut in this network. Moreover, there exists a flow of maximum size which is integral.*

Let us also note the following nearly obvious fact.

**Observation 62.** *If a network  $(\vec{G}, s, t)$  contains a flow  $f$  of non-zero size, then  $\vec{G}$  contains a directed path from  $s$  to  $t$  using only edges  $e$  such that  $f(e) \neq 0$ .*

*Proof.* Let  $S$  be the set of all vertices reachable from  $s$  by a directed path using only edges  $e$  such that  $f(e) \neq 0$ . If  $t \in S$ , then  $\vec{G}$  contains a directed path from  $s$  to  $t$  using only edges  $e$  such that  $f(e) \neq 0$ , and we are done.

Suppose for a contradiction that  $t \notin S$ . Then

$$0 < m = \partial_f s = \sum_{v \in S} \partial_f v = - \sum_{e=(x,y) \in E(\vec{G}): x \in S, y \notin S} f(e) \leq 0;$$

indeed, the contributions of the edges between vertices of  $S$  to this sum cancel out, and  $f(e) = 0$  for all edges  $e = (x, y)$  with  $x \in S$  and  $y \notin S$  by the definition of  $S$  (otherwise the vertex  $y$  would be reachable in  $\vec{G}_f$  by a directed path using only edges with non-zero flow value, and thus it would belong to  $S$ ). This is a contradiction.  $\square$

From this, it is easy to see that integral flows can be decomposed to edge-disjoint paths from  $s$  to  $t$ .

**Lemma 63.** *Let  $(\vec{G}, s, t)$  be a network and let  $f$  be an integral flow of size  $m$  in this network. Then  $\vec{G}$  contains  $m$  pairwise edge-disjoint directed paths from  $s$  to  $t$  containing only edges  $e$  such that  $f(e) = 1$ .*

*Proof.* We proceed by induction on  $m$ . If  $m = 0$ , then there is nothing to prove, and thus assume that  $m > 0$  and that the claim holds for all flows of smaller size. By Observation 62, there exists a directed path  $\vec{P}_m$  from  $s$  to  $t$  using only edges to which  $f$  assigns value 1. Let  $f'$  be obtained from  $f$  by decreasing the values on the edges of the path  $\vec{P}_m$  to 0; then  $f'$  clearly is a flow of size  $m - 1$  in  $(\vec{G}, s, t)$ . By the induction hypothesis,  $\vec{G}$  contains  $m - 1$  pairwise edge-disjoint paths  $\vec{P}_1, \dots, \vec{P}_{m-1}$  from  $s$  to  $t$  using only edges to which  $f'$  assigns value 1. These paths are clearly edge-disjoint from  $\vec{P}_m$ .  $\square$

We can now directly translate this to our undirected setting.

**Theorem 64** (Edge version of Menger's theorem, fixed ends). *Let  $G$  be a graph and let  $u$  and  $v$  be distinct vertices of  $G$ . For any positive integer  $k$ , exactly one of the following claims holds: Either*

- (a)  $G$  contains at least  $k$  pairwise edge-disjoint paths from  $u$  to  $v$ , or

(b) there exists an edge-cut in  $G$  of size less than  $k$  separating  $u$  from  $v$ .

*Proof.* Clearly (a) and (b) cannot hold at the same time, since in case (a), each of the  $k$  paths from  $u$  to  $v$  has to cross each edge-cut separating  $u$  from  $v$ , and thus each such edge-cut has to have size at least  $k$ .

Consider the network  $(\vec{G}, u, v)$ , where  $\vec{G}$  is obtained from  $G$  by replacing each edge  $xy$  by a pair of oppositely directed edges  $(x, y)$  and  $(y, x)$ . By Theorem 61, there exists a cut  $S$  of capacity  $m$  and an integral flow  $f$  of the same size in this network.

Let  $X$  be the set of edges of  $G$  with one end in  $S$  and the other end in  $V(G) \setminus S$ . Since  $u \in S$  and  $v \notin S$ , note that  $X$  is an edge-cut in  $G$  separating  $u$  from  $v$ . Moreover,  $|X| = m$ . Hence, if  $m < k$ , then (b) holds.

Therefore, suppose that  $m \geq k$ . Without loss of generality, we can assume that for every edge  $xy \in E(G)$ , we have  $f(x, y) = 0$  or  $f(y, x) = 0$ ; indeed, if  $f(x, y) = f(y, x) = 1$ , we can decrease the amount of flow on both  $(x, y)$  and  $(y, x)$  to 0, obtaining a flow of the same size. By Lemma 63, there exists a system of pairwise edge-disjoint directed paths in  $\vec{G}$  from  $u$  to  $v$ . Then the corresponding undirected paths between  $u$  and  $v$  in  $G$  are also pairwise edge-disjoint, and there are  $m \geq k$  of them. Therefore, (a) holds.  $\square$

We can now easily characterize  $k$ -edge-connectivity.

**Theorem 65** (Edge version of Menger's theorem). *For every graph  $G$  and positive integer  $k$ , the following claims are equivalent:*

- (i) The graph  $G$  is  $k$ -edge-connected.
- (ii) For all distinct vertices  $u, v \in V(G)$ , there exist  $k$  pairwise edge-disjoint paths in  $G$  from  $u$  to  $v$ .

*Proof.* If  $G$  has only one vertex, then both (i) and (ii) trivially hold. Hence, we can assume that  $|V(G)| \geq 2$ .

Suppose first that (ii) holds, and consider any edge-cut  $X$  in  $G$ . The graph  $G - X$  is not connected, and thus there exist vertices  $u$  and  $v$  contained in different components of  $G - X$ . By (ii), there exist  $k$  pairwise edge-disjoint paths in  $G$  from  $u$  to  $v$ , and each of these paths must intersect  $X$  in at least one edge. Hence,  $|X| \geq k$ . Thus shows that every edge-cut in  $G$  has size at least  $k$ , and thus  $G$  is  $k$ -edge-connected, i.e., (i) holds.

Conversely, suppose that (i) holds, and consider any distinct vertices  $u$  and  $v$  in  $G$ . Since  $G$  is  $k$ -connected, it does not contain any edge-cut of size less than  $k$ , and thus the claim (b) from the statement of Theorem 64 is false. Therefore, (a) has to hold and  $G$  contains  $k$  pairwise edge-disjoint paths from  $u$  to  $v$ . Since this is the case for all distinct  $u, v \in V(G)$ , the claim (ii) holds.  $\square$

## 10.2 Vertex connectivity

For the vertex version of the notion, we essentially want to proceed similarly, defining the vertex-connectivity as the maximum  $k$  such that the removal of

any at most  $k - 1$  vertices does not disconnect the graph. There is a small issue here: it is not possible to disconnect a clique by removing any number of vertices. Thus, we need to do something special for cliques. As a warning, not everyone agrees what is the right thing to do, so many texts define the connectivity for cliques differently.

For a non-negative integer  $k$ , we say that a graph  $G$  is  $k$ -connected if the graph  $G - X$  is connected for every set  $X \subseteq V(G)$  of size less than  $k$ , and we define the *connectivity*  $\kappa(G)$  of  $G$  to be the largest non-negative integer  $k \leq |V(G)| - 1$  such that  $G$  is  $k$ -connected. Thus, the clique  $K_n$  is  $k$ -connected for every integer  $k$ , but its connectivity is  $n - 1$ . A set  $X \subseteq V(G)$  is a *cut* if  $G - X$  is not connected; thus, a graph is  $k$ -connected if and only if all cuts in  $G$  have size at least  $k$ .

It seems easier to disconnect the graph by deleting vertices than edges; indeed, an edge cut  $X$  can usually be turned into at most as large vertex cut  $X'$  by taking one end of each edge of  $X$ . We need to be a bit careful, though: What if  $G - X$  has only two components, and  $X'$  contains all vertices of one of them? Still, with a bit of care, we can prove the following (see the tutorials).

**Observation 66.** *Every graph  $G$  satisfies  $\kappa(G) \leq \lambda(G)$ .*

We would now like to prove a vertex version of Menger's theorem. It is possible to get one for paths between pairs of vertices similar to Theorem 64 (as we are going to discuss in the tutorials), but it is somewhat simpler to look for paths between sets of vertices. For a graph  $G$  and sets  $A, B \subseteq V(G)$ , an  $(A, B)$ -linkage is a system of pairwise vertex-disjoint paths in  $G$ , each with one end in  $A$  and the other end in  $B$ . Note that an  $(A, B)$ -linkage may contain also paths consisting just of a single vertex (when  $A \cap B \neq \emptyset$ ).

**Theorem 67** (Menger's theorem, fixed ends). *Let  $G$  be a graph and let  $A, B \subseteq V(G)$  be subsets of its vertex set. For every positive integer  $k$ , exactly one of the following claims holds: Either*

- (a)  $G$  contains an  $(A, B)$ -linkage of size at least  $k$ , or
- (b) there exists a set  $X \subseteq V(G)$  of size less than  $k$  such that every path from  $A$  to  $B$  in  $G$  intersects  $X$ .

*Proof.* Clearly (a) and (b) cannot hold at the same time, and thus it suffices to show that at least one of them holds.

We turn  $G$  into a directed graph by splitting each vertex into an "input" and "output" part, as follows. Let  $\vec{G}$  be the directed graph with vertex set  $\{s, t\} \cup \{v^i, v^o : v \in V(G)\}$  and the following edges:

- $(v^i, v^o)$  for each  $v \in V(G)$ ,
- $(u^o, v^i)$  and  $(v^o, u^i)$  for each edge  $uv \in E(G)$ ,
- $(s, v^i)$  for each  $v \in A$ , and
- $(v^o, t)$  for each  $v \in B$ .

Let  $f$  be a flow in the network  $(\vec{G}, s, t)$  of maximum size  $m$ , without loss of generality integral.

Suppose first that  $m \geq k$ . By Lemma 63, there exist a system  $\vec{\mathcal{P}}$  of  $m$  pairwise edge-disjoint paths in  $\vec{G}$  from  $s$  to  $t$ . Consider any path  $\vec{P} = sv_1^o v_1^i v_2^o v_2^i \dots v_k^o v_k^i t \in \vec{\mathcal{P}}$ , and note that  $P = v_1 v_2 \dots v_k$  is a path in  $G$  from  $A$  to  $B$ . Moreover, note that for distinct directed paths  $\vec{P}_1, \vec{P}_2 \in \vec{\mathcal{P}}$ , the paths  $P_1$  and  $P_2$  are vertex-disjoint: If they intersected in a vertex  $v$ , then both  $\vec{P}_1$  and  $\vec{P}_2$  would contain the edge  $(v^i, v^o)$ . Consequently, the system  $\{P : \vec{P} \in \vec{\mathcal{P}}\}$  is an  $(A, B)$ -linkage of size  $m \geq k$  in  $G$ , and (a) holds.

Conversely, suppose that  $m < k$ . By Theorem 61, the network  $(\vec{G}, s, t)$  contains a cut  $S$  of capacity less than  $k$ . Let us choose such a cut with  $|S|$  maximal. Then for every edge  $uv \in E(G)$ ,

$$\text{if } u^o \in S, \text{ then } v^i \in S. \quad (10.1)$$

Indeed, if  $v^i \notin S$ , then consider the cut  $S' = S \cup \{v^i\}$ . The capacity of the cut  $S'$  is at most as large as the capacity of the cut  $S$ , since

- the edge  $u^o v^i$  contributes to the capacity of  $S$  but not to the capacity of  $S'$ , and
- only one edge, namely the edge  $(v^i, v^o)$ , can contribute to the capacity of  $S'$  but not to the capacity of  $S$ .

Since  $|S'| > |S|$ , this would contradict the choice of  $S$ ; and this contradiction shows that (10.1) holds.

Let  $X = \{v \in V(G) : v^i \in S, v^o \notin S\} \cup \{v \in A : v^i \notin S\} \cup \{v \in B : v^o \in S\}$ . For each vertex  $v \in X$ , the edge  $(v^i, v^o)$  or the edge  $(s, v^i)$  or the edge  $(v^o, t)$  contributes to the capacity of the cut  $S$ , and thus  $|X| \leq |S| < k$ . We claim that every every path from  $A$  to  $B$  in  $G$  intersects  $X$ , and thus (b) holds. Indeed, suppose for a contradiction that there exists a path  $P$  from a vertex  $u \in A$  to a vertex  $v \in B$  in  $G - X$ . Since  $u, v \notin X$ , we have  $u^i \in S, v^o \notin S$ , and  $v^i \notin S$ . Let  $z$  be the first vertex of  $P$  such that  $z^i \notin S$ , and let  $y$  be its predecessor in the path  $P$ ; thus  $y^i \in S$ . Since  $y \notin X$ , we have  $y^o \in S$ . But then (10.1) implies that  $z^i \in S$ , which is a contradiction.  $\square$

We can now get a connectivity version of this theorem.

**Theorem 68** (Menger's theorem). *Let  $G$  be a graph. For every positive integer  $k$ , the following claims are equivalent:*

- (i) *The graph  $G$  is  $k$ -connected.*
- (ii) *For all sets  $A, B \subseteq V(G)$  such that  $|A| = |B| \leq k$ , there exists an  $(A, B)$ -linkage of size  $|A|$  in  $G$ .*

*Proof.* Suppose that (i) holds, and consider any sets  $A, B \subseteq V(G)$  of the same size  $m \leq k$ . If  $X \subseteq V(G)$  has size less than  $m$ , then the  $k$ -connectivity of  $G$  implies that  $G - X$  is connected, and since  $|A| = |B| = m > |X|$ , the graph

$G - X$  contains a vertex of  $A$  and a vertex of  $B$ . Thus,  $G$  contains a path from  $A$  to  $B$  disjoint from  $X$ . It follows that (b) of Theorem 67 is false, and thus (a) holds, i.e.,  $G$  contains an  $(A, B)$ -linkage of size  $m$ . Therefore, (ii) holds.

Conversely, suppose that (ii) holds. We claim that  $G - X$  is connected for every set  $X \subseteq V(G)$  of size less than  $k$ , and thus (i) holds. Indeed, otherwise we can choose vertices  $u$  and  $v$  belonging to different components of  $G - X$  and let  $A = X \cup \{u\}$  and  $B = X \cup \{v\}$ . We have  $|A| = |B| = |X| + 1 \leq k$ , but we cannot find  $|X| + 1$  pairwise vertex-disjoint paths from  $A$  to  $B$ , since all such paths must intersect  $X$ .  $\square$

### 10.3 Homework

1. Let  $G$  be a graph with at least two vertices, let  $v$  be a vertex of  $G$ , and let  $e$  be an edge of  $G$ . Show that

$$\lambda(G) - 1 \leq \lambda(G - e) \leq \lambda(G)$$

$$\kappa(G) - 1 \leq \kappa(G - e) \leq \kappa(G)$$

$$\kappa(G) - 1 \leq \kappa(G - v)$$

On the other hand, show that

- there exists a graph  $G$  and a vertex  $v \in V(G)$  such that  $\lambda(G) \geq 10^6$  and  $\lambda(G - v) = 0$ , and
  - there exists a graph  $G$  and a vertex  $v \in V(G)$  such that  $\lambda(G) = \kappa(G) = 0$  and  $\lambda(G - v) \geq 10^6$ .
2. Let  $G$  be a graph and let  $k$  be a positive integer. For vertices  $u, v \in V(G)$ , we write  $u \sim v$  if  $u = v$  or  $G$  contains at least  $k$  pairwise edge-disjoint paths from  $u$  to  $v$ . Show that the relation  $\sim$  is an equivalence. Hint: It will be useful to prove that if  $u \sim v \sim w$ , then every edge-cut in  $G$  separating  $u$  from  $w$  has size at least  $k$ .
  3. Let  $G$  be a graph of maximum degree at most three. Show that for every positive integer  $k$ , if  $G$  is  $k$ -edge-connected, then  $G$  is  $k$ -connected. Hint: For contradiction, suppose that  $G$  contains a cut of size less than  $k$ , and show you can turn it into an edge-cut of size less than  $k$  using the assumption that all vertices have degree at most three.

## 10.4 Tutorial

1. Prove that if a network  $(\vec{G}, s, t)$  contains a flow  $f$  of non-zero size, then  $\vec{G}$  contains a directed path from  $s$  to  $t$  using only edges  $e$  such that  $f(e) \neq 0$ .
2. Determine the edge-connectivity and the connectivity of the complete bipartite graph  $K_{n,m}$ .
3. Show that
  - a graph  $G$  contains an edge-cut of size at most  $k$  if and only if we can partition the vertices of  $G$  into two non-empty disjoint parts  $A$  and  $B$  so that  $G$  contains at most  $k$  edges with one end in  $A$  and the other end in  $B$ ; and
  - a graph  $G$  contains a cut of size at most  $k$  if and only if we can partition the vertices of  $G$  into three parts  $A$ ,  $B$ , and  $X$ , where  $A$  and  $B$  are non-empty,  $G$  does not contain any edges between  $A$  and  $B$ , and  $|X| \leq k$ .
4. Show that a connected graph  $G$  is 2-edge-connected if and only if every edge of  $G$  is contained in a cycle.
5. Show that every graph  $G$  satisfies  $\kappa(G) \leq \lambda(G)$ .
6. Let  $G$  be a graph and let  $u$  and  $v$  be distinct vertices of  $G$ . Show that  $G$  contains  $\kappa(G)$  distinct paths from  $u$  to  $v$  pairwise vertex-disjoint except for their shared ends. Hint: Apply a theorem from the lecture to obtain a linkage between closed neighborhoods of  $u$  and  $v$ .

## Lesson 11

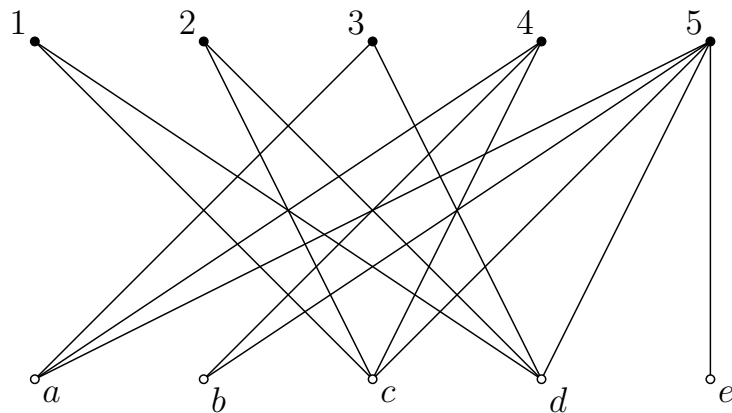
# Matchings in bipartite graphs

### 11.1 Perfect matchings in bipartite graphs

Suppose that a teacher wants to assign one of five essay topics to each of the five students in his class. He wants each topic to be assigned to only one student, and he also wants to take the preferences of the students into account: Each student should work on a topic that they like. Suppose that

- the 1-st student likes topics  $c$  and  $d$ ,
- the 2-nd student also likes topics  $c$  and  $d$ ,
- the 3-rd student likes topics  $a$  and  $d$ ,
- the 4-th student likes the topics  $a$ ,  $b$ , and  $c$ , and
- the 5-th student likes all the topics.

The situation is depicted in the following picture:



Does an assignment of topics to students satisfying these properties exist?

Note that this would change if the 3-rd student changed their mind and decided they no longer like the topic  $a$ . If that happened, no such assignment would exist: Indeed, then there would be only two topics ( $c$  and  $d$ ) liked by at least one of the first three students, and thus there would not be enough topics to assign to them.

It turns out that a similar reasoning can always be applied to solve this kind of problems. To see that, let us start with some definitions. A *matching* in a graph  $G$  is any 1-regular subgraph  $M$  of  $G$ . We say that  $M$  *covers* a set  $S \subseteq V(G)$  of vertices if each vertex of  $S$  is incident with an edge belonging to  $M$ , i.e., if  $S \subseteq V(M)$ . We say that a matching  $M$  is *perfect* if it covers all vertices of  $G$ , i.e., if  $V(M) = V(G)$ .

In the problem we considered at the beginning of the lecture, we were interested in finding a perfect matching between the students and the topics in the graph representing the preferences of the students. In this case, the graph was bipartite (one part corresponded to the students, the other part to the topics, and each edge had one end in the first part and the other end in the second one). While there exists a polynomial-time algorithm to find a perfect matching in a general graph, it is rather complicated. In this lecture, we are going to only consider matchings in bipartite graphs, where the situation is rather simpler.

Let  $G$  be a bipartite graph with parts  $A$  and  $B$ . For a set  $S \subseteq A$ , let  $N(S)$  denote the set of vertices in  $B$  with at least one neighbor in  $S$ .

Clearly, if  $G$  has a perfect matching (or even a matching covering  $A$ ), then  $|N(S)| \geq |S|$  holds for every  $S \subseteq A$ , since all vertices matched to those in  $S$  must belong to  $N(S)$ . The following important result tells us that this necessary condition is also sufficient.

**Theorem 69** (Hall's theorem). *A bipartite graph  $G$  with parts  $A$  and  $B$  has a matching which covers  $A$  if and only if  $|N(S)| \geq |S|$  holds for every set  $S \subseteq A$ .*

*Proof.* We have already argued that this condition is necessary, and thus we only need to show that it is sufficient. That is, suppose that  $|N(S)| \geq |S|$  holds for every set  $S \subseteq A$ ; we need to show that  $G$  contains a matching which covers  $A$ . Consider any set  $X \subseteq V(G)$  of size less than  $|A|$ , and let  $S = A \setminus X$ . We have

$$\begin{aligned} |N(S)| &\geq |S| = |A \setminus X| = |A| - |A \cap X| = |A| - (|X| - |B \cap X|) \\ &= |B \cap X| + |A| - |X| > |B \cap X|, \end{aligned}$$

and thus there exists a vertex  $v \in N(S) \setminus X$ . Since  $v \in N(S)$ , there exists an edge  $uv$  with  $u \in S$ . Note that  $u \in A \setminus X$  and  $v \in B \setminus X$ .

Therefore, we have shown that for every set  $X \subseteq V(G)$  of size less than  $|A|$ ,  $G$  contains a path (of length one) from  $A$  to  $B$  disjoint from  $X$ . By Menger's theorem from the previous lecture,  $G$  contains  $|A|$  pairwise vertex-disjoint paths from  $A$  to  $B$ . Clearly each of these paths consists of only one edge, and thus their union is a matching in  $G$  covering  $A$ .  $\square$

With this, we can easily characterize the bipartite graphs that have a perfect matching.

**Corollary 70.** *A bipartite graph  $G$  with parts  $A$  and  $B$  has a perfect matching if and only if  $|A| = |B|$  and  $|N(S)| \geq |S|$  holds for every set  $S \subseteq A$ .*

*Proof.* A matching that covers  $A$  is perfect if and only if  $|B| = |A|$ ; the rest follows from Hall's theorem.  $\square$

Let us remark that Theorem 69 can be also proved by considering flows in networks. Indeed, let  $(\vec{G}, s, t)$  be the network obtained from  $G$  as follows: We direct all edges of  $G$  from  $A$  to  $B$ , add a vertex  $s$  and edges from  $s$  to all vertices of  $A$ , and add a vertex  $t$  and edges from all vertices of  $B$  to  $t$ . It is easy to see that  $G$  has a matching which covers  $A$  if and only if there exists a flow of size  $|A|$  in the network  $(\vec{G}, s, t)$ . Hall's theorem then can be proved by analysing the capacity of cuts in this network. An advantage of this approach is that this directly gives you a polynomial-time algorithm to find the perfect matching (or decide it does not exist), which you have likely seen before in Algorithms and Data Structures 1 class.

Let us note that Hall's theorem has the following nice consequence.

**Corollary 71.** *For every  $d \geq 1$ , every  $d$ -regular bipartite graph  $G$  has a perfect matching.*

*Proof.* Let  $A$  and  $B$  be the parts of  $G$ . Note that  $d|A| = |E(G)| = d|B|$ , since each vertex in  $A$  is incident with  $d$  edges, each vertex in  $B$  is incident with  $d$  edges, and each edge has one end in each of  $A$  and  $B$ . It follows that  $|A| = |B|$ .

Next, consider any set  $S \subseteq A$  and let  $Y$  be the set of edges of  $G$  with an end in  $S$ . On one hand, each vertex of  $S$  is incident with  $d$  edges, and thus  $|Y| = d|S|$ . On the other hand, each edge from  $Y$  has an end in  $N(S)$  and each vertex of  $N(S)$  is incident with at most  $d$  edges belonging to  $Y$  (as well as possibly with additional edges not incident with  $S$ ), and thus  $|Y| \leq d|N(S)|$ . This gives  $|N(S)| \geq |Y|/d = |S|$ .

Since  $|A| = |B|$  and  $|N(S)| \geq |S|$  for every set  $S \subseteq A$ , Corollary 70 implies that  $G$  has a perfect matching.  $\square$

## 11.2 Matchings with preferences

Let us now consider a variation on our initial example: What if the students do not like/dislike all topics to the same extent, but have more nuanced preferences? One possible solution would be to ask each student to assign a numeric preference to each topic according to how much they like them, e.g., an integer between 0 ("I hate this topic") and 10 ("I love this one"), and then find a matching that maximizes the sum of preferences. This leads to the *maximum weight bipartite matching* problem, which can be formally stated as follows: Given two sets  $A$  and  $B$  and a weight function  $w : A \times B \rightarrow \mathbb{R}$ , find an injective function

$f : A \rightarrow B$  maximizing  $\sum_{a \in A} w(a, f(a))$ . This problem can be solved in polynomial time using linear programming or other specialized algorithms, and this is the case even for the analogous problem in general (possibly non-bipartite) graphs.

However, this may not be quite what we want. With numeric preferences, there is always the difficulty that people may use different scales to assign their values, and thus summing the preferences does not necessarily make sense. As another option, we could ask the students just to order the topics from their most to their least favorite one, and try to select an assignment based on this information. Of course, in this situation it is not quite clear what the best solution actually means. E.g., if everyone orders the topics in the same way, there will always be the one person who gets assigned their least favorite topic, who will hardly consider the solution to be fair.

To help us here, let us also specify the preferences on the other side of the bipartite graph. E.g., for each topic, the teacher can give an ordering of the students based on how qualified they are to work on it. We can then hope to use this ordering to “break the ties” somehow.

Of course, it is still not clear which matchings between the students and topics to prefer; however, a situation that we definitely would like to avoid is the following one: Suppose that the student 1 is assigned the topic  $y$  and the student 2 is assigned the topic  $x$ , but the student 1 would much prefer the topic  $x$  and they are also more qualified to deal with it than the student 2. In this situation, the student 1 would certainly have a reason to feel that the assignment is not quite fair.

Formally, for two sets  $A$  and  $B$  of the same size, a *system of preferences* is a system  $\{\succ_x : x \in A \cap B\}$ , where  $\succ_a$  for  $a \in A$  is a linear ordering of  $B$  and  $\succ_b$  for  $b \in B$  is a linear ordering of  $A$ . As before, we represent a matching between  $A$  and  $B$  by a bijective function  $f : A \rightarrow B$ . We say that a matching  $f$  is *stable* if for every  $a \in A$  and  $b \in B$  such that  $f(a) \neq b$ , we have  $f(a) \succ_a b$  or  $f^{-1}(b) \succ_b a$ . E.g., in our example, the student  $a$  can always find a reason why they were not assigned the topic  $b$ : Either they actually prefer the topic  $f(a)$  they got assigned, or the student  $f^{-1}(b)$  chosen to deal with the topic  $b$  is more qualified to do it than the student  $a$ .

Let us remark that the question of finding a stable matching is sometimes called the *stable marriage problem*; I will leave an interpretation of this naming up to your imagination.

Of course, it is not at all clear that a stable matching always exists, or how to find it. The following result addresses both of these issues.

**Theorem 72.** *Let  $A$  and  $B$  be two sets of the same size  $n$ . For every system  $\{\succ_x : x \in A \cap B\}$  of preferences, there exists a stable matching; moreover, such a matching can be found in polynomial time.*

*Proof.* Let us consider the following algorithm (by Gale and Shapley). Throughout the algorithm, we *provisionally match* some elements of  $A$  to elements of  $B$  (initially, no elements are provisionally matched). Moreover, for each element

$a \in A$ , we maintain a list  $L_a \subseteq B$  of *potential matches* for  $a$ , initially consisting of all of  $B$ .

The algorithm proceeds in rounds. In each round, we do the following:

- Every element  $a \in A$  which is not provisionally matched to anyone picks and removes the most preferred element from  $L_a$  (according to the ordering  $>_a$ ). If there is no one to pick, i.e.,  $L_a$  is empty at the beginning of this round, we ignore  $a$  (this will actually never happen, but allowing for this possibility simplifies the analysis of the algorithm).
- For each element  $b \in B$  which was picked by at least one element of  $A$  in this round, let  $s_b$  be the most preferred element that picked it (according to the ordering  $>_b$ ). If  $b$  is not yet provisionally matched, or if they prefer  $s_b$  over the element  $a_b \in A$  they are currently provisionally matched with, then we provisionally match  $s_b$  with  $b$  (and unmatch  $a_b$  if it exists).

We repeat this until no element is picked in the first part of the current round. In each of the previous rounds, we have picked at least one element and permanently removed it from the list of potential matches of an element of  $A$ . Since the lists of potential matches initially contain  $n^2$  elements in total, the algorithm necessarily stops after at most  $n^2$  rounds.

We claim that all elements become provisionally matched in the end. Otherwise, since  $|A| = |B|$ , there would exist unmatched elements  $a \in A$  and  $b \in B$ . Note that if  $b$  were picked in any round, then it would be provisionally matched to an element of  $A$ , and would stay provisionally matched till the end (possibly to different elements of  $A$ ). Hence,  $b$  was never picked, and in particular the element  $a$  never picked  $b$ . However, then  $b$  must still be contained in the list  $L_a$ . This is a contradiction, since then the algorithm would not stop yet.

Let  $f : A \rightarrow B$  be the function assigning to each element  $a \in A$  the element with which it is provisionally matched at the end of the algorithm. We claim that the matching  $f$  is stable. Indeed, consider any element  $b \in B$  different from  $f(a)$ . If  $a$  prefers  $f(a)$  over  $b$ , then the condition of stability for this pair is satisfied. Hence, suppose that  $a$  prefers  $b$  over  $f(a)$ , i.e.,  $b >_a f(a)$ . The element  $a$  picked  $f(a)$  from  $L_a$  in the round in which it got provisionally matched to  $f(a)$ . Since it prefers  $b$ , it had to pick and remove  $b$  from  $L_a$  in some earlier round. However, observe that the element  $f^{-1}(b)$  is the most preferred element of  $A$  (according to  $>_b$ ) that picked  $b$  at some point in the algorithm. This implies that  $f^{-1}(b) >_b a$ , and thus the condition of stability for this pair is again satisfied.  $\square$

Of course, we can ask additional questions. In general, there can exist many stable matchings; which one of them do we want to choose? Note that the Gale-Shapley algorithm is not symmetric with respect to the exchange of  $A$  and  $B$ , and it turns out to bias in favour of the elements of  $A$ : Among all stable matchings, each element of  $A$  will end up matched to the best possible element of  $B$ , while each element of  $B$  will end up matched to the worst possible element of  $A$  (according to their preferences); this may or may not be what we want.

Furthermore, from the practical perspective, we need to consider whether it is in everyone's best interest to be honest, or if someone may get a better result by lying about its preferences. There is a whole branch of algorithmic game theory studying this kind of questions, in case you are interested.

### 11.3 Homework

1. You have a deck of cards with 4 suits and 13 values per suit (52 cards in total). You shuffle it in any way and deal 13 different piles, each pile containing 4 cards. Show that you can always select exactly one card from each pile so that the 13 selected cards have all 13 different values. Hint: Consider the bipartite graph joining each value to all piles that contain a card of this value. Given a subset of values, how many piles must (at least) contain cards whose value belongs to this subset?
2. Let  $d$  be a non-negative integer. Show that a graph  $G$  has an orientation such that every vertex has outdegree at most  $d$  if and only if every subgraph  $F$  of  $G$  has at most  $d|V(F)|$  edges. Hint: Consider matchings in the bipartite graph where one side is formed by the edges of  $G$ , the other side contains  $d$  copies of each vertex of  $G$ , and each edge  $uv$  is adjacent to all copies of  $u$  and  $v$ .
3. Use the previous exercise to show that every planar graph has an orientation such that every vertex has outdegree at most three.

## 11.4 Tutorial

1. Find a perfect matching in the following graphs, or show that they do not have one:

TODO

2. For a graph  $G$ , let  $\mu(G)$  denote the maximum size (number of edges) of a matching in  $G$ . Let  $\nu(G)$  denote the minimum size of a set  $X \subseteq V(G)$  such that every edge of  $G$  has at least one end in  $X$ . Show that  $\nu(G) = |V(G)| - \alpha(G)$  and determine  $\mu(K_n)$ ,  $\nu(K_n)$ ,  $\mu(K_{n,m})$ , and  $\nu(K_{n,m})$ .
3. Show that

$$\mu(G) \leq \nu(G) \leq 2\mu(G)$$

holds for every graph  $G$ . Hint: Consider a largest matching  $M$  in  $G$ . What can you say about the set  $V(G) \setminus V(M)$ ?

4. Let  $G$  be a bipartite graph with parts  $A$  and  $B$  and let

$$r = \max_{S \subseteq A} (|S| - |N(S)|).$$

Show that  $G$  has a matching which covers  $A$  if and only if  $r = 0$ , and that  $\mu(G) = |A| - r$ . Hint: for the second part, add  $r$  new vertices adjacent to all vertices of  $A$  and show that the resulting modified graph has a matching that covers  $A$ .

5. Show that every bipartite graph  $G$  satisfies  $\nu(G) = \mu(G)$ .
6. Describe an algorithm based on maximum flows in networks to compute  $\mu(G)$  for a bipartite graph  $G$ . Show that using this algorithm, you can find a largest independent set in  $G$  in polynomial time.

## Lesson 12

# Expanders

While (edge-)connectivity is an important graph parameter, it does not actually work particularly well as a measure of reliability of a network. E.g., consider the graph consisting of a clique and of a single vertex  $v$  adjacent to only one vertex of the clique. The edge-connectivity and connectivity of this graph is 1, i.e., from the perspective of these parameters, the corresponding network does not seem to be very robust. However, unless you happen to be the owner of the computer represented by vertex  $v$ , the network will actually be extremely reliable for you.

What might be a better notion? Given any set  $S$  of computers (vertices of the network), there always is a way to make sure they cannot communicate—just break all the computers in this set  $S$ . We might say that the network is reliable if there is no substantially simpler way to disconnect any set  $S$  of vertices. This is one of motivations for the following definitions.

Given a graph  $G$  and a set  $S$  of its vertices, we define  $N_G(S)$  as the set of vertices  $v \in V(G) \setminus S$  such that  $v$  has at least one neighbor in  $S$ ; and we define  $\partial_G S$  to be the set of edges of  $G$  with one end in  $S$  and the other end in  $V(G) \setminus S$ . For a positive real number  $a$ , we say that  $G$  is

- an *a-expander* if  $|N_G(S)| \geq a|S|$  holds for every set  $S \subset V(G)$  of size at most  $\frac{1}{2}|V(G)|$ , and
- an *a-edge-expander* if  $|\partial_G(S)| \geq a|S|$  holds for every set  $S \subset V(G)$  of size at most  $\frac{1}{2}|V(G)|$ .

### 12.1 Basic properties of expanders

Let us start with an easy observation, which follows from the fact that each vertex of  $N_G(S)$  is incident with an edge belonging to  $\partial_G S$ .

**Observation 73.** *If a graph is an a-expander, it is also an a-edge-expander.*

Note that the restriction to subsets containing at most half of the vertices is quite natural (e.g., we of course cannot take  $S = V(G)$ , as there would be no vertices outside of  $S$ ). It also is just enough of a restriction to imply many interesting properties. For example, the *diameter* of a graph is the maximum distance between its vertices; and it is easy to check that expanders must have diameter at most logarithmic in the number of their vertices.

**Lemma 74.** *For every positive real number  $a$  and for every  $n$ -vertex graph  $G$ , if  $G$  is an  $a$ -expander, then  $G$  has diameter at most  $2 + 2\lceil \log_{1+a} \frac{n}{2} \rceil$ .*

*Proof.* For a vertex  $v \in V(G)$  and a non-negative integer  $r$ , let  $N_r[v]$  denote the set of vertices of  $G$  at distance at most  $r$  from  $v$ . Note that  $N_{r+1}[v] = N_r[v] \cup N_G(N_r[v])$ , and thus if  $|N_r[v]| \leq \frac{n}{2}$ , then  $|N_{r+1}[v]| \geq (1+a)|N_r[v]|$ . By induction, this implies that for every  $r \geq 0$ , either  $|N_r[v]| > \frac{n}{2}$  or  $|N_r[v]| \geq (1+a)^r$ ; and thus for  $r_0 = 1 + \lceil \log_{1+a} \frac{n}{2} \rceil$ , we have  $|N_{r_0}[v]| > \frac{n}{2}$ .

Hence, if we consider any vertices  $u, v \in V(G)$ , we have  $|N_{r_0}[u]| + |N_{r_0}[v]| > n$ , and thus there exists a vertex  $z \in N_{r_0}[u] \cap N_{r_0}[v]$ . Then the distance from  $z$  to each of  $u$  and  $v$  is at most  $r_0$ , and by the triangle inequality, the distance between  $u$  and  $v$  is at most  $2r_0$ .  $\square$

It is not particularly hard to find expanders when we allow the degrees of the vertices to be arbitrarily large. Moreover, in practical applications, we are typically interested in expanders  $G$  with bounded maximum degree  $\Delta$ . In this case, it does not matter much whether we consider expanders or edge-expanders; the following observation holds, because each vertex of  $N_G(S)$  is incident with at most  $\Delta$  edges belonging to  $\partial_G S$ , and thus  $|N_G(S)| \geq \frac{1}{\Delta} |\partial_G S|$  holds for every set  $S$  of vertices of  $G$ .

**Observation 75.** *If a graph of maximum degree  $\Delta$  is an  $a$ -edge-expander, then it is also an  $\frac{a}{\Delta}$ -expander.*

It is not hard to see that for any  $d \geq 3$ , a random  $d$ -regular graph is almost surely an expander. For a basic idea, consider the following lemma, where we show the existence of a related object. A *one-sided*  $(n, m, \Delta, \beta, \gamma)$ -expander is a bipartite graph  $G$  with parts  $A$  and  $B$  of sizes  $n$  and  $m$ , respectively, such that all vertices in  $A$  have degree at most  $\Delta$  and such that  $|N_G(S)| > (\Delta - \beta)|S|$  holds for every non-empty set  $S \subseteq A$  of size at most  $\gamma n$ .

**Lemma 76.** *Let  $A$  and  $B$  be sets of vertices of size  $n$  and  $m$ , respectively, and let  $d \geq 3$  be an integer. Let  $G$  be the random graph obtained as follows: For each vertex  $v \in A$ , we add edges to  $d$  (not necessarily distinct) vertices of  $B$  selected independently uniformly at random. Then  $G$  is a one-sided  $(n, m, d, 2, \gamma)$ -expander, where  $\gamma = \frac{m^2}{2e^{d-1}(d-2)^2 n^2}$ .*

*Proof.* For a set  $S \subseteq A$  of size  $s$ , the probability that  $|N_G(S)| \leq (d-2)s$  is at most

$$\binom{m}{(d-2)s} \cdot \left( \frac{(d-2)s}{m} \right)^{ds} \leq \left( \frac{em}{(d-2)s} \right)^{(d-2)s} \left( \frac{(d-2)s}{m} \right)^{ds} = \left( \frac{e^{d-2}(d-2)^2 s^2}{m^2} \right)^s.$$

Consider any positive integer  $s \leq \gamma n$ . The expected number of sets  $S \subseteq A$  of size  $s$  such that  $|N_G(S)| \leq (d-2)s$  is at most

$$\binom{n}{s} \cdot \left( \frac{e^{d-2}(d-2)^2 s^2}{m^2} \right)^s \leq \left( \frac{en}{s} \right)^s \cdot \left( \frac{e^{d-2}(d-2)^2 s^2}{m^2} \right)^s = \left( \frac{e^{d-1}(d-2)^2 n}{m^2} \cdot s \right)^s \\ \left( \frac{e^{d-1}(d-2)^2 n}{m^2} \cdot \gamma n \right)^s \leq 2^{-s}.$$

Consequently, the expected number of non-empty sets  $S \subseteq A$  of size at most  $\gamma n$  such that  $|N_G(S)| \leq (d-2)|S|$  is at most  $\sum_{s=1}^{\lfloor \gamma n \rfloor} 2^{-s} < 1$ , and thus there is no such set with non-zero probability.  $\square$

When  $m = n$ , this shows that subsets of  $A$  of size up to  $\frac{1}{2e^{d-1}(d-2)} \cdot n$  have excellent expansion properties. A similar argument gives that sets of size up to  $n$  also have reasonably good expansion properties (though of course not by as good factor as  $d-2$ ). We have not considered the subsets of  $B$  (the construction is not symmetric with respect to exchange of  $A$  and  $B$ ), and of course  $G$  is not  $d$ -regular. The actual argument in the setting of  $d$ -regular graphs (in the configuration model) is a bit more technical.

It is rather more difficult to construct expanders with bounded degrees deterministically, but nowadays this problem is well understood, and we know many different constructions. Their analysis is rather involved and beyond the scope of this lecture; however, we can at least introduce an important tool used in this context.

## 12.2 Spectral expanders

It turns out that the edge-expansion properties of a graph are closely related to its algebraic properties. For a graph  $G$  with vertex set  $\{z_1, \dots, z_n\}$ , the *Laplacian matrix* of  $G$  is the  $n \times n$  matrix  $L$  such that

$$L_{i,j} = \begin{cases} -1 & \text{if } i \neq j \text{ and } z_i z_j \in E(G) \\ 0 & \text{if } i \neq j \text{ and } z_i z_j \notin E(G) \\ \deg z_i & \text{if } i = j. \end{cases}$$

That is,  $L$  is obtained from the adjacency matrix of  $G$  by negating it and putting the degrees of the vertices of  $G$  on the diagonal.

Let us now consider any set  $S \subseteq V(G)$  and let  $1_S$  be its *characteristic vector*, i.e., the column vector whose  $i$ -th coordinate is 1 if  $z_i \in S$  and 0 otherwise. Consider any vertex  $z_i \in S$ ; the value of the  $i$ -th entry of the vector  $L1_S$  is  $\deg z_i -$  the number of neighbors of  $z_i$  in  $S$ , that is, the number of edges of  $\partial_G S$  incident with  $z_i$ . Consequently, we obtain the following useful fact:

$$|\partial_G S| = 1_S^T L 1_S. \quad (12.1)$$

To analyze this product, it will be convenient to use the eigenvalues. We need a result from linear algebra which states that for every symmetric matrix, we can choose an orthonormal basis formed by its eigenvectors.

**Theorem 77.** For every symmetric  $n \times n$  matrix  $A$ , there exist an orthonormal basis  $v_1, \dots, v_n$  and real numbers  $\lambda_1, \dots, \lambda_n$  such that  $Av_i = \lambda_i v_i$  holds for every  $i \in \{1, \dots, n\}$ .

Let us recall that  $v_1, \dots, v_n$  is an *orthonormal basis* if  $v_i^T v_j = 0$  for all distinct  $i, j \in \{1, \dots, n\}$  and  $v_i^T v_i = 1$  for each  $i \in \{1, \dots, n\}$ . Note that the Laplacian matrix  $L$  has one obvious eigenvector, namely the vector  $j$  whose entries are all equal to 1; we have  $Lj = 0$ . Theorem 77 actually holds in a slightly stronger form: We can pick an arbitrary eigenvector of norm 1 for each distinct eigenvalue to the basis. Thus, we can assume that  $v_1 = \frac{1}{\sqrt{n}}j$  and  $\lambda_1 = 0$  (the factor  $\frac{1}{\sqrt{n}}$  is chosen so that  $v_1^T v_1 = 1$ ). Moreover, note the following observation.

**Observation 78.** All eigenvalues of the Laplacian matrix of a graph are non-negative.

*Proof.* Let  $L$  be the Laplacian matrix of a graph  $G$  with vertex set  $\{z_1, \dots, z_n\}$  and let  $\lambda$  be an eigenvalue of  $L$ . Let  $v$  be the corresponding eigenvector, that is,  $v$  is a non-zero vector such that  $Lv = \lambda v$ . We can assume that  $v$  has a positive entry (otherwise, we can consider the vector  $-v$  instead of  $v$ ). Let  $i \in \{1, \dots, n\}$  be an index such that the entry  $v_i$  of  $v$  is maximum possible; we have  $v_i > 0$ . Note that

$$(Lv)_i = v_i \deg z_i - \sum_{j: z_i z_j \in E(G)} v_j = \sum_{j: z_i z_j \in E(G)} (v_i - v_j) \geq 0.$$

On the other hand, since  $Lv = \lambda v$ , we also have  $(Lv)_i = \lambda v_i$ . Therefore,  $\lambda v_i \geq 0$ , and since  $v_i > 0$ , this implies that  $\lambda \geq 0$ .  $\square$

Using these observations, we can prove the following important result.

**Theorem 79.** Let  $L$  be the Laplacian matrix of a graph  $G$ , let  $v_1 = \frac{1}{\sqrt{n}}j$ ,  $v_2, \dots, v_n$  be an orthonormal basis formed by eigenvectors of  $L$ , and let  $\lambda_1 = 0, \lambda_2, \dots, \lambda_n$  be the corresponding eigenvalues. Let  $\lambda = \min(\lambda_2, \dots, \lambda_n)$ . Then  $G$  is a  $\frac{\lambda}{2}$ -edge-expander.

*Proof.* Consider any set  $S \subset V(G)$  such that  $|S| \leq \frac{n}{2}$ . Since  $v_1, \dots, v_n$  is a basis, we have  $1_S = \sum_{i=1}^n a_i v_i$  for some real numbers  $a_1, \dots, a_n$ . By (12.1), we have

$$\begin{aligned} |\partial_G S| &= 1_S^T L 1_S = \left( \sum_{i=1}^n a_i v_i \right)^T L \left( \sum_{i=1}^n a_i v_i \right) \\ &= \sum_{i,j=1}^n a_i a_j v_i^T L v_j = \sum_{i,j=1}^n a_i a_j \lambda_j v_i^T v_j \\ &= \sum_{i=1}^n \lambda_i a_i^2 = \sum_{i=2}^n \lambda_i a_i^2. \end{aligned}$$

Similarly, we have

$$|S| = \mathbf{1}_S^T \mathbf{1}_S = \sum_{i=1}^n a_i^2.$$

We can also express  $|S|$  in another way:

$$|S| = \mathbf{j}^T \mathbf{1}_S = \sqrt{n} \cdot \mathbf{v}_1^T \mathbf{1}_S = \sqrt{n} \cdot \sum_{i=1}^n a_i \mathbf{v}_1^T \mathbf{v}_i = \sqrt{n} \cdot a_1.$$

Therefore,

$$\sum_{i=2}^n a_i^2 = |S| - a_1^2 = |S| - \frac{|S|^2}{n} = |S| \cdot \frac{n - |S|}{n} \geq \frac{|S|}{2}.$$

It follows that

$$\frac{|\partial_G S|}{S} = \frac{\sum_{i=2}^n \lambda_i a_i^2}{|S|} \geq \frac{\sum_{i=2}^n \lambda_i a_i^2}{2 \sum_{i=2}^n a_i^2} \geq \frac{\lambda}{2}.$$

□

Motivated by this result, for a positive real number  $\lambda$  we say that a graph  $G$  with Laplacian matrix  $L$  is a  $\lambda$ -spectral-expander if the multiplicity of 0 as an eigenvalue of  $L$  is 1 and all non-zero eigenvalues of  $L$  are greater or equal to  $\lambda$ . Thus, Theorem 79 shows that every  $\lambda$ -spectral-expander is a  $\frac{\lambda}{2}$ -edge-expander.

Interestingly, a rough converse holds as well.

**Theorem 80.** *For a positive real number  $a$ , if a graph  $G$  of maximum degree  $\Delta$  is an  $a$ -edge-expander, then it is also a  $\frac{a^2}{2\Delta}$ -spectral-expander.*

Thus, for a graph of bounded maximum degree, we can (at least approximately) determine how good expander it is by examining the eigenvalues of its Laplacian matrix.

## 12.3 Expander codes

Expanders have many applications throughout discrete mathematics and computer science. Importantly, they in many aspects behave as random graphs, even though they may be constructed through completely deterministic means. Thus, they are often useful as explicit examples of graphs with certain properties. They are used in design of parallelized algorithms (e.g., sorting networks of logarithmic depth) to ensure efficient spread of the information among the processing units. They are also used in derandomization, to obtain deterministic algorithms from randomized ones. Here, we describe a simple application in design of error-correcting codes. The key observation this construction uses is as follows.

**Lemma 81.** *Let  $G$  be bipartite graph with parts  $A$  and  $B$ . If  $G$  is a one-sided  $(n, m, \Delta, \beta, \gamma)$ -expander with  $2\beta \leq \Delta$ , then for every non-empty set  $S \subseteq A$  of size at most  $2(1 - \beta/\Delta)(\gamma n - 1)$ , there exists a vertex  $u \in B$  with exactly one neighbor in  $S$ .*

*Proof.* Let  $S_0$  be a subset of  $S$  of size  $\min(|S|, \lfloor \gamma n \rfloor)$ ; then  $|N_G(S_0)| > (\Delta - \beta)|S_0|$ . Moreover, since  $|S| \leq 2(1 - \beta/\Delta)(\gamma n - 1)$  and  $2\beta \leq \Delta$ , we have

$$\begin{aligned} \Delta|S| &\leq 2(\Delta - \beta)(\gamma n - 1) \leq 2(\Delta - \beta)\lfloor \gamma n \rfloor \text{ and} \\ \Delta|S| &\leq 2(\Delta - \beta)|S|, \end{aligned}$$

and thus  $\Delta|S| \leq 2(\Delta - \beta)|S_0|$ . Let  $X$  be a set of size  $|N_G(S_0)|$  containing exactly one edge from each vertex of  $N_G(S_0)$  to  $S_0$ . Note that  $G$  contains at most

$$\Delta|S| - |X| \leq \Delta|S| - (\Delta - \beta)|S_0| \leq (\Delta - \beta)|S_0| < |N_G(S_0)|$$

edges from  $S$  to  $N_G(S_0)$  not belonging to  $X$ . Therefore, there exists a vertex  $u \in N_G(S_0)$  such that there exists only one edge from  $u$  to  $S$ , namely the one belonging to  $X$ .  $\square$

Let  $M$  be the bipartite adjacency matrix of such a graph  $G$ ; more precisely, the rows of  $M$  are indexed by  $B$ , columns by  $A$ , and  $M_{u,v} = 1$  if  $uv \in E(G)$  and  $M_{u,v} = 0$  otherwise. Let us now consider the linear code with check matrix  $M$ . Recall that this code is an  $(n, n - m, d)$ -code, where  $d$  is the minimum number of columns of  $M$  that sum to 0. Observe that columns with indices in a set  $S \subseteq A$  sum to 0 if and only if every vertex of  $B$  has even number of neighbors in  $S$  (and in particular no vertex of  $B$  has exactly one neighbor in  $S$ ). By Lemma 81, this shows that  $d > 2(1 - \beta/\Delta)(\gamma n - 1)$ .

Thus, these codes have rate  $1 - \frac{m}{n}$  and relative distance roughly  $2(1 - \beta/\Delta)\gamma$ . As we can see from Lemma 76, this provides codes with both rate and relative distance greater than some fixed positive constant.

## 12.4 Tutorial

1. Show that if  $G$  is an  $n$ -vertex graph of maximum degree at most two, then  $G$  is not an  $a$ -expander for any  $a > \frac{4}{n-1}$ .
2. Let  $a$  be a positive real number. Show that if an  $n$ -vertex graph  $G$  is an  $a$ -expander, then for every set  $X \subseteq V(G)$  of size less than  $\frac{an}{4}$ , the graph  $G - X$  has a component of size at least  $n - (1 + 1/a)|X|$ . Hint: Consider the union  $S$  of the vertex sets of the components of  $G - X$  except for the largest one. If  $|S| > n/2$ , instead show that we can choose some components of  $G - X$  so that the union  $S'$  of their vertex sets satisfies  $n/4 \leq |S'| \leq n/2$ .
3. Let  $L$  be the Laplacian matrix of an  $n$ -vertex graph  $G$ , let  $v_1 = \frac{1}{\sqrt{n}}j$ ,  $v_2, \dots, v_n$  be an orthonormal basis formed by eigenvectors of  $L$ , and let  $\lambda_1 = 0, \lambda_2, \dots, \lambda_n$  be the corresponding eigenvalues. Show that  $G$  is connected if and only if the eigenvalues  $\lambda_2, \dots, \lambda_n$  are non-zero. Hint: Modify the proof of the observation that all eigenvalues of the Laplacian matrix of a graph are non-negative to show that if  $Lv = 0$  for a non-zero vector  $v$ , then there exists a component  $C$  of  $G$  such that all entries of  $v$  corresponding to the vertices of  $C$  have the same value.
4. Let  $G$  be a  $d$ -regular graph, let  $L$  be its Laplacian matrix, and let  $\lambda$  be an eigenvalue of  $L$ . Show that  $\lambda \leq 2d$ , and that  $2d$  is an eigenvalue of  $L$  if and only if  $G$  is bipartite.
5. Let  $G$  be a one-sided  $(n, n, d, \beta, \gamma)$ -expander of maximum degree at most  $d$ , and let  $A$  and  $B$  be the parts of  $G$ . Let  $M_1, \dots, M_d$  be pairwise edge-disjoint matchings in  $G$  such that  $E(G) = E(M_1) \cup \dots \cup E(M_d)$  (as a bonus task, you can try to show that such a system of matchings exists in every bipartite graph of maximum degree at most  $d$ ).

Suppose we are given real numbers  $t_1, \dots, t_{2n}$  and we run the following algorithm: Initially, we assign these numbers to distinct vertices of  $G$  arbitrarily. We then perform  $d$  rounds. In the  $r$ -th round, for each edge  $ab \in E(M_r)$ , where  $a \in A$  and  $b \in B$ , if the number currently assigned to  $a$  is greater than the one assigned to  $b$ , we exchange the two numbers.

Note that this algorithm can be performed in time  $O(d)$  using  $O(n)$  processors in parallel, since for each of the matchings, we can process the edges independently. Show that after we run this algorithm, for every positive integer  $k \leq (d - \beta + 1)(\gamma n - 1)$ , all but at most  $\frac{1}{d - \beta + 1}k$  of the  $k$  largest elements of the set  $\{t_1, \dots, t_{2n}\}$  are assigned to vertices belonging to  $B$ . Hint: For contradiction, consider a set  $S$  of  $\lfloor \frac{1}{d - \beta + 1}k \rfloor + 1$  vertices of  $A$  to which some of the  $k$  largest elements would be assigned. What can we say about the numbers assigned to the vertices in  $N_G(S)$ ?

## Lesson 13

# Sublinear separators in planar graphs

Can planar graphs be good expanders? If we go over a few examples, it seems that this is not the case. E.g., in a  $k \times k$  grid  $G$ , there exists a set  $S$  containing roughly half of the vertices (the left half of the grid) with  $|N_G(S)| = k$ , showing that it is not more than a  $\frac{2}{k}$ -expander. It turns out that this is essentially the best possible.

To see this, let us start by proving a variant of Menger's theorem for planar graphs. We first need to argue that the following seemingly unrelated claim holds. Consider a triangulation (a plane graph whose faces are triangles) with its vertices colored (not necessarily properly) by three colors. A face of this triangulation is *rainbow* if all three colors appear on its vertices.

**Lemma 82.** *For every coloring of vertices of a triangulation  $G$  by three colors, the number of rainbow faces is even.*

*Proof.* We can assume that the colors are 1, 2, and 3. Let  $H$  be the auxiliary graph whose vertices are faces of  $G$  and two vertices  $f_1$  and  $f_2$  of  $H$  are adjacent if and only if the faces  $f_1$  and  $f_2$  share an edge whose ends have colors 1 and 2. Observe that a vertex  $f$  of  $H$  has degree one when the face  $f$  of  $G$  is rainbow, degree two if  $f$  is incident with a vertex of color 1 and two vertices of color 2 or a vertex of color 2 and two vertices of color 1, and zero otherwise.

Therefore,  $H$  is a disjoint union of isolated vertices, cycles, and paths, and the rainbow faces of  $G$  are exactly the ends of paths of  $H$ . Hence,  $G$  has an even number of rainbow faces.  $\square$

This has the following consequence. A graph  $G$  drawn in the plane is an *internal triangulation* if all internal faces of  $G$  are triangles and the outer face is bounded by a cycle (of any length).

**Lemma 83.** *Let  $G$  be an internal triangulation with the outer face bounded by a cycle  $C$ . Let  $L$  and  $R$  be vertex-disjoint subpaths of  $C$  and let  $T$  and  $B$  be the*

two subpaths of  $C$  edge-disjoint from  $L$  and  $R$  such that  $C = L \cup T \cup R \cup B$ . For any set  $X \subseteq V(G)$ , either  $G[X]$  contains a path from  $V(T)$  to  $V(B)$ , or  $G - X$  contains a path from  $V(L)$  to  $V(R)$ .

*Proof.* Let us extend the graph  $G$  to a triangulation  $G'$  by adding a vertex  $x$  adjacent to all vertices of  $T$ , a vertex  $y$  adjacent to all vertices of  $L$ , a vertex  $z$  adjacent to all vertices of  $R \cup B$ , and the edges of the triangle  $xyz$ . Let  $X' = X \cup \{x\}$ . Let us color all vertices of the component of  $G'[X']$  containing  $x$  by color 1, all vertices of the component of  $G' - (X' \cup \{z\})$  containing  $y$  by color 2, and all other vertices by color 3. The face  $xyz$  is rainbow, and thus by Lemma 82, there exists another rainbow face  $uvw$  of  $G'$ ; say  $u$  has color 1,  $v$  has color 2, and  $w$  has color 3. Note that  $w$  is adjacent both to a vertex belonging to the the component of  $G'[X']$  containing  $x$  and to a vertex belonging to the component of  $G' - (X' \cup \{z\})$  containing  $y$ , but belongs to neither of these components; this is only possible if  $w = z$ . But then the edge  $uv$  belongs either to the path  $R$  or to the path  $B$ . In the former case,  $G - X$  contains a path from  $V(L)$  to  $v \in V(R)$ , and in the latter case,  $G[X]$  contains a path from  $V(T)$  to  $u \in V(B)$ .  $\square$

With this, we can now give the promised variant of Menger's theorem for planar graphs.

**Theorem 84.** *Let  $G$  be an internal triangulation with the outer face bounded by a cycle  $C$  and let  $L$  and  $R$  be vertex-disjoint subpaths of  $C$  with the same number  $k$  of vertices. Let  $T$  and  $B$  be the two subpaths of  $C$  edge-disjoint from  $L$  and  $R$  such that  $C = L \cup T \cup R \cup B$ . Then exactly one of the following claims holds: Either*

- (a)  $G$  contains  $k$  pairwise vertex-disjoint paths from  $V(L)$  to  $V(R)$ , or
- (b) there exists a path  $P$  in  $G$  from  $V(T)$  to  $V(B)$  with less than  $k$  vertices.

*Proof.* If (b) holds, then by planarity every path from  $V(L)$  to  $V(R)$  in  $G$  must pass through  $V(P)$ , and thus (a) cannot hold.

If (a) does not hold, then by Menger's theorem, there exists a set  $X \subset V(G)$  of size less than  $k$  such that every path from  $V(L)$  to  $V(R)$  in  $G$  passes through  $X$ . By Lemma 83,  $G[X]$  contains a path from  $V(T)$  to  $V(B)$ . This path has at most  $|X| < k$  vertices, and thus (b) holds.  $\square$

Consider a cycle  $C$  in a graph  $G$ . A *shortcut* is a path in  $G$  with distinct ends  $u$  and  $v$  on  $C$  and otherwise disjoint from  $C$  such that  $L$  is shorter than both paths between  $u$  and  $v$  in  $C$ . We say that a cycle is *geodesic* if there do not exist any shortcuts. Theorem 84 implies that planar triangulations cannot contain too long geodesic cycles compared to their size.

**Corollary 85.** *Let  $k$  be a positive integer and let  $G$  be an internal triangulation with the outer face bounded by a cycle  $C$  of length at least  $4k$ . If the cycle  $C$  is geodesic, then  $|V(G)| \geq k^2$ .*

*Proof.* Since  $C$  has length at least  $4k$ , we can split it into four pairwise edge-disjoint paths  $L, T, R$ , and  $B$ , each with at least  $k$  vertices. Since  $C$  is geodesic,  $G$  does not contain a path from  $V(T)$  to  $V(B)$  with less than  $k$  vertices. By Theorem 84,  $G$  contains  $k$  pairwise vertex-disjoint paths  $P_1, \dots, P_k$  from  $V(L)$  to  $V(R)$ . By a symmetric argument,  $G$  also contains  $k$  pairwise vertex-disjoint paths  $Q_1, \dots, Q_k$  from  $V(T)$  to  $V(B)$ . For every  $i, j \in \{1, \dots, k\}$ , the paths  $P_i$  and  $Q_j$  intersect in at least one vertex  $v_{i,j}$ . Note that if  $i', j' \in \{1, \dots, k\}$  are indices such that  $(i, j) \neq (i', j')$ , then  $v_{i',j'} \neq v_{i,j}$  (if  $i' \neq i$ , then  $v_{i',j'}$  does not lie on the path  $P_i$ , and if  $j' \neq j$ , then it does not lie on the path  $Q_j$ ). Therefore,  $G$  has at least  $k^2$  distinct vertices.  $\square$

Given a cycle  $C$  in a plane graph  $G$ , let  $\text{In}(C)$  be the set of vertices drawn in the region of the plane bounded by  $C$  (excluding the vertices of  $C$  itself). Let  $\text{Out}(C)$  be the vertices drawn strictly outside of this region, again excluding the vertices of  $C$ . Thus,  $\text{In}(C)$ ,  $V(C)$ , and  $\text{Out}(C)$  is a partition of the vertex set of  $G$ . Moreover, the planar drawing ensures that  $G$  does not contain any edges between  $\text{In}(C)$  and  $\text{Out}(C)$ . Let us now prove the key result, showing that every triangulation contains a cycle dividing it into parts of roughly equal size.

**Theorem 86.** *In every triangulation  $G$  with  $n$  vertices, there exists a cycle  $C$  of length at most  $4\lceil\sqrt{n+1}\rceil$  such that  $|\text{In}(C)| \leq \frac{2}{3}n$  and  $|\text{Out}(C)| \leq \frac{2}{3}n$ .*

*Proof.* Let  $k = \lceil\sqrt{n+1}\rceil$ . We say that a cycle  $C$  in  $G$  is *promising* if  $C$  has length at most  $4k$  and  $|\text{Out}(C)| \leq \frac{2}{3}n$ . Note that  $G$  contains at least one promising cycle, namely the triangle bounding the outer face of  $G$ . Let us choose a promising cycle  $C$  for which  $|\text{Out}(C)|$  is largest possible, and subject to that  $C$  is longest possible.

Let  $H$  be the subgraph of  $G$  drawn in the closed region bounded by  $C$ ; thus,  $H$  is an internal triangulation with the outer face bounded by  $C$ . We claim that the cycle  $C$  is not geodesic in  $H$ . Indeed, suppose for a contradiction that it is geodesic, and in particular induced. Let  $uvw$  be any internal face of  $H$  sharing an edge  $uw$  with  $C$ , and let  $C' = (C - uw) \cup uvw$ . Note that  $C'$  is a cycle in  $G$ ,  $\text{Out}(C') = \text{Out}(C)$  and  $|C'| > |C|$ . By the choice of  $C$ , the cycle  $C'$  cannot be promising, and thus  $|C'| > 4k$ . This implies that  $|C| = k$ . However, then Corollary 85 implies that  $|V(H)| \geq k^2 > n$ , which is a contradiction.

Therefore,  $H$  contains a shortcut  $P$  of  $C$ . Let  $C_1$  and  $C_2$  be the two cycles in  $C \cup P$  different from  $C$ . Since  $P$  is a shortcut, we have  $|C_1|, |C_2| < |C| \leq 4k$ . Moreover,  $|\text{Out}(C_1)|, |\text{Out}(C_2)| > |\text{Out}(C)|$ , since for  $i \in \{1, 2\}$ , we have  $\text{Out}(C_i) = \text{Out}(C) \cup (V(C) \setminus V(C_i))$ , and  $V(C_i) \neq V(C)$  since  $P$  is a shortcut. By the maximality of  $|\text{Out}(C)|$ , it follows that neither  $C_1$  nor  $C_2$  can be a promising cycle, and thus  $|\text{Out}(C_1)| > \frac{2}{3}n$  and  $|\text{Out}(C_2)| > \frac{2}{3}n$ . Then for  $i \in \{1, 2\}$ , we have  $|\text{In}(C_i)| = |V(G) \setminus (V(C_i) \cup \text{Out}(C_i))| = n - |C_i| - |\text{Out}(C_i)| < \frac{1}{3}n - |C_i|$ . Consequently,

$$|\text{In}(C)| = |\text{In}(C_1)| + |\text{In}(C_2)| + |V(P)| - 2 < \frac{2}{3}n - |C_1| - |C_2| + |V(P)| - 2 < \frac{2}{3}n,$$

and  $C$  has all properties required by the statement of the theorem.  $\square$

Since every planar graph can be extended to a triangulation by adding edges (and this only makes the graph harder to separate), this has the following consequence.

**Corollary 87.** *The vertex set of every planar graph  $G$  with  $n$  vertices can be partitioned into parts  $I$ ,  $C$ , and  $O$  so that  $|I|, |O| \leq \frac{2}{3}n$ ,  $|C| \leq 4\lceil\sqrt{n+1}\rceil$ , and  $G$  does not have any edges between  $I$  and  $O$ .*

Note that  $I$  and  $O$  also cannot be too small; indeed,  $|I| \geq n - |C| - |O| \geq \frac{1}{3}n - 4\lceil\sqrt{n+1}\rceil = \Omega(n)$ , and similarly  $|O| = \Omega(n)$ . This then implies that planar graphs indeed cannot be good expanders: Since  $|I| + |O| \leq n$ , we can without loss of generality assume  $|I| \leq \frac{1}{2}$ , and

$$\frac{|N_G(I)|}{|I|} \leq \frac{|C|}{|I|} = O\left(\frac{1}{\sqrt{n}}\right).$$

Corollary 87 makes it possible to improve the efficiency of algorithms for various problems, using the divide-and-conquer method. For that, an algorithmic version of this corollary is needed: It is not sufficient to know that the small separator  $C$  exists, but we also need to be able to find it efficiently. This is not hard to do; indeed, we can turn the proof of Theorem 86 to an algorithm returning the cycle  $C$ , with time complexity  $O(n^2)$ .

As an example application, suppose that we want to find the largest independent set in an  $n$ -vertex planar graph  $G$ . In general graphs, the best algorithms for this problem have exponential time complexity,  $\exp(\Theta(n))$ , and there are argument supporting the idea that this is the best possible.

For planar graphs, we can use the following approach: We find a partition of the vertex set of  $G$  to parts  $I$ ,  $C$ , and  $O$  as in Corollary 87. We then go over all independent sets  $A$  of  $G[C]$ , and we check which of them extends to a largest possible independent set in  $G$  by adding vertices not in  $C$ . Since there are no edges between  $I$  and  $O$ , a largest independent set in  $G$  extending  $A$  consists of any largest independent set in  $G[I] - N_G(A)$ , the set  $A$ , and any largest independent set in  $G[O] - N_G(A)$ . We find largest independent sets in the planar subgraphs  $G[I] - N_G(A)$  and  $G[O] - N_G(A)$  recursively. What is the time complexity  $t(n)$  of this algorithm? There are at most  $2^{|C|} = 2^{O(\sqrt{n})}$  choices for the set  $A$ , and for each of them, we recurse on two subgraphs, each of which has size at most  $\frac{2}{3}n$ . This gives the following recurrence (valid say for  $n \geq 1$ , with the basic case  $t(x) = 1$  for  $x \leq 1$ ):

$$t(n) \leq O(n^2) + 2^{O(\sqrt{n})}t\left(\frac{2}{3}n\right).$$

Since  $O(n^2)$  is much smaller than  $2^{O(\sqrt{n})}$ , we can ignore this term and obtain a simpler recurrence

$$t(n) \leq 2^{O(\sqrt{n})}t\left(\frac{2}{3}n\right).$$

By repeatedly using this recurrence, we get

$$\begin{aligned} t(n) &\leq 2^{O(\sqrt{n})} \cdot 2^{O(\sqrt{(2/3)n})} \cdot 2^{O(\sqrt{(2/3)^2n})} \dots \\ &\leq \exp\left(O\left(\sum_{i \geq 0} \sqrt{(2/3)^i n}\right)\right) = \exp\left(O\left(\sqrt{n} \sum_{i \geq 0} (\sqrt{2/3})^i\right)\right) \\ &= \exp(O(\sqrt{n})). \end{aligned}$$

Thus, while the time complexity of this algorithm is not polynomial, it is bounded by a function growing much slower than the  $\exp(\Theta(n))$  exponential one.