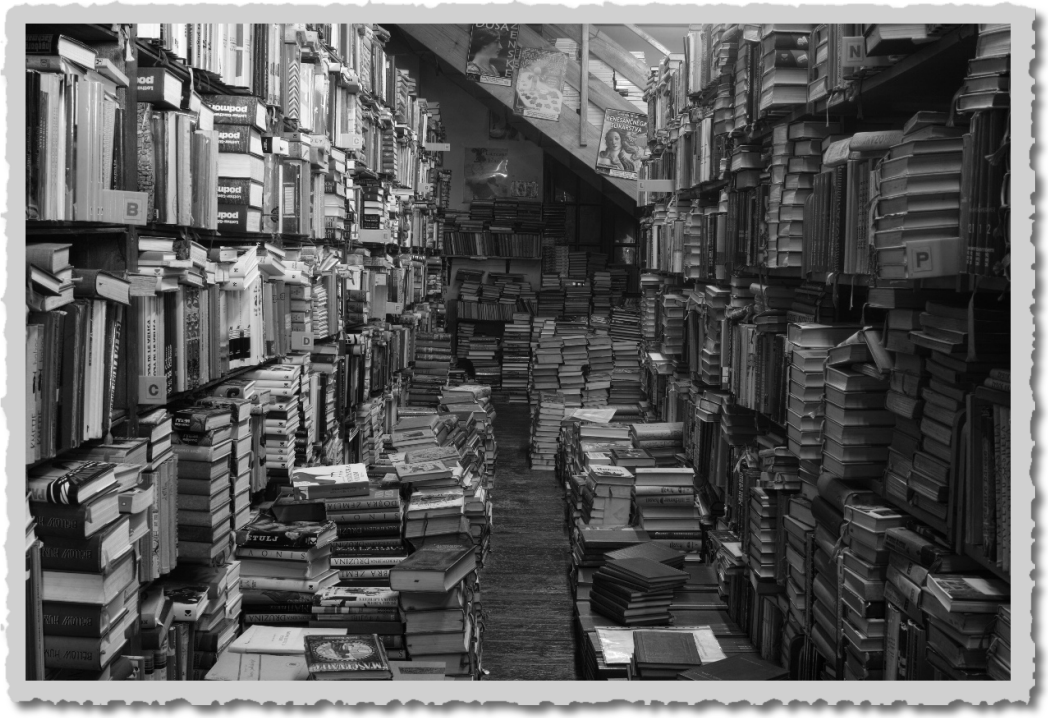


Interesting Topics for Bachelor Theses

Walter M. Böhm
Institute for Statistics and Mathematics
Vienna University of Economics
walter.boehm@wu.ac.at

October 4, 2018

The picture on the title page is an artwork by Jakob Böhm,
www.jacob-julian.com

The other pictures used in this book are from Wikipedia Commons and MacTutor History of Mathematics archive maintained at The University of St. Andrews.

Foreword

This booklet is a collection of topics which I prepared over the time for my students. The selection of topics reflects my personal interests and is therefore biased towards combinatorial mathematics, probability and statistics, operations research, scheduling theory and, yes, history of mathematics.

It is in the nature of things that the level of difficulty varies from topic to topic. Some are technically more demanding others somewhat easier. The only prerequisite to master these topics are courses in mathematics and statistics at an undergraduate university level. Otherwise, no special prior knowledge in mathematics is afforded.

However, what is needed is *serious interest in mathematics*, of course.

How is a Topic organized?

Each topic consists of three parts:

(1) *An Invitation*

Of course, the major purpose of this invitation is to raise your interest and to draw your attention to a problem which I found very interesting, attractive and challenging. Further, in each invitation I introduce some basic terminology so that you can start reading basic literature related to the topic.

(2) *Where to go from here*

Some of my invitations are more detailed depending on the topic, so you may ask yourself: *Is there anything left for me to do?*

Yes, there is lot of work still to be done. The second section of each topic contains questions and problems which you may study in your thesis. This list is by no means exhaustive, so there enough opportunity to unleash your creative potential and hone your skills. For some topics I explicitly indicate some issues of general interest, these are points which you should to discuss in your thesis in order to make it more or less self-contained and appealing to readers not spezialized in this topic. And sometimes there is also a section *What to be avoided*: here I indicate aspects and issues related to the topic which may lead too far afield or are technically too difficult

(3) *An annotated bibliography*

This is a commented list of interesting, helpful and important books and journal articles.

This book has not been finished yet and probably may never be.

Contents

Topic 1

Recreational Mathematics

A Contradiction in Terms?

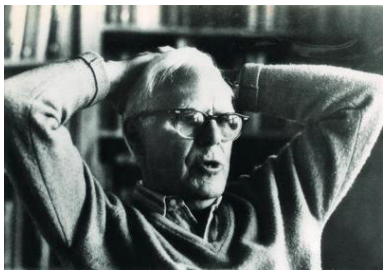
Mathematics is too serious and, therefore, no opportunity should be missed to make it amusing.

Blaise Pascal[1]

Keywords: *exciting puzzles, mathematical riddles and mysteries*
recreational mathematics

1.1 An Invitation

The subtitle of this topic suggests that mathematics and recreation do not fit together nicely. Mathematics is generally considered a hard and dry business, how can that be reconciled with enjoyable activities like relaxation and recreation?



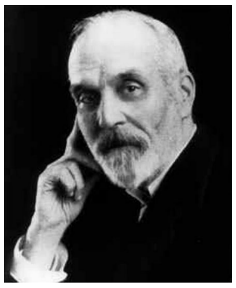
Martin Gardner
1914–2010

Martin Gardner (1959b) writes in the foreword of one of his wonderful books: It's the element of *play* which makes recreational mathematics recreational and this may take many forms, may it be solving a puzzle, a magic trick, a paradox, a fallacy, an exciting game. And it's the delight and intellectual pleasure we experience when having solved a difficult puzzle. For this reason it should not come as a surprise that even most brilliant scientists could not resist the temptations of recreational mathematics. Indeed, being a connoisseur of puzzles and the like you are in best company: Visitors of Albert Einstein, for instance, reported that in his bookshelf he always had a section stocked with mathematical puzzles and games.

[1] Cited from Petkovic (2010).

1.1.1 The Challenge of Mathematical Puzzles

Mathematical puzzles are a passion of mine since my childhood and I never missed an opportunity to get hands on an apparently new one. Unfortunately, some fifty years ago those opportunities were rarer than they are today. Now we have the world wide web and finding new and interesting puzzles is very easy. But in those days one had to rely mostly on newspapers and magazines. Some of them, the better ones, had puzzle corners in their weekend editions. There you could find besides the obligatory big weekend crossword various picture puzzles, also called *rebus*, and even problems coming with a mathematical flavor. Rebus were often nothing more than pictures cut at random and one had to rearrange the snippets properly, a very boring business, recommended only to feeble minded persons, as Henry E. Dudeney[2], once remarked. But from time to time one could find nice little gems, wonderful mathematical puzzles, exciting challenges of your mind. Higher mathematical education or scholarship was usually not required to solve these newspaper puzzles, but originality, diligence, patience and some basic understanding of logic were very helpful. Solutions of these puzzles were given only one week later in the next weekend edition. So, either you solved the puzzle and trusted in your solution, or you had to wait patiently. In the meantime one could discuss the problem with classmates, friends and even teachers.



Henry E. Dudeney
1857–1930

Others at my age collected stamps, I collected interesting puzzles by cutting them out of the newspapers and storing the clippings in folders which, as years passed by, grew bigger and bigger. Eventually it became necessary to bring some order into my collection. So I began to categorize my puzzles into those belonging to arithmetics and geometry, number theory, logic, combinatorics, magic squares, graph theory and probability. All these are very renown fields of mathematics, though frankly speaking, at an age of fourteen my mathematical knowledge was a *quantité négligeable*. But this changed by and by because by solving puzzles I learned quite a lot. Well, not necessarily *useful* mathematics in the sense that the acquired knowledge was of much use in our math lessons in school. For instance, knowing how to construct a magic square will be of no help when dealing with problems from elementary analysis.

Over time I also realized that there exist special *books* solely devoted to recreational mathematics and that some of these books were available in public libraries in Vienna. This opened up a whole new world when I could read the wonderful books of H. E. Dudeney and Martin Gardner, collections of most exciting and challenging mathematical puzzles and games.

What makes mathematical puzzles so attractive, not only to me but to so many other people? Certainly, it is *problem solving*, it's the excitement when working

[2]Henry E. Dudeney, the famous British grand master of puzzles and mathematical games.

on a puzzle, it's the intellectual pleasure when having solved it. Interestingly, problem solving in the realm of recreational mathematics is not of highest esteem among many professional mathematicians, particularly those adhering to the pure doctrine. An exemplary representative of these was *Edmund Landau*[3]. He once coined the somewhat contemptuous term *Schmierölmathematik*. This is certainly a fairly extreme view of matters. Indeed, when you perform scholarly research in the literature on recreational mathematics, you will soon find out that creative mathematicians are seldom ashamed about their interest in recreational topics. And regarding problem solving: *Andrei A. Markov*[4], once remarked (Basharin, Langville, and Naumov, 2004):

> *The alleged opinion that studies in classes are of the highest scientific nature, while exercises in solving problems are of lowest rank, is unfair. Mathematics to a considerable extent consists in solving problems and together with proper discussion, this can be of the highest scientific nature while studies in classes might be of the lowest rank.*

There's nothing more to be said, I think, except that recreational mathematics is pure mathematics uncontaminated by utility (copyright Martin Gardner (1959b)).

While problem solving lies at the heart of mathematical puzzles, it would be certainly wrong to classify as puzzle whenever a mathematical problem is solved, often by extensive and complicate reasoning and calculation, think of hard exam problems, for instance.

What we need, is some kind of working definition: *What is a mathematical puzzle?*

It's surprisingly difficult to arrive at a definition which finds general acceptance[5]. But there are some distinguishing characteristics of mathematical puzzles Peter Winkler (2011) has worked out when reviewing the book by Miodrag Petkovic (2010) on famous puzzles of great mathematicians.

- First and foremost: A puzzle is an engaging, self-contained mathematical question.
- A puzzle should have a *raison d'être*, something that makes it worth thinking about.
- It should be easy to communicate among people.
- No special devices like high speed computers are required, all you need is paper and pencil, if at all.

We may take these characteristics as cornerstones of a more complete and acceptable definition. But for the moment, let's dispense with formalities, let's look for some recreation and diversion. What would be better suited for this purpose than some fine puzzle?

[3] Edmund Landau, 1877–1938, German mathematician.

[4] Andrei A. Markov, 1856–1922, Russian mathematician, see *Topic 4: The Chains of Andrei A. Markov* on page 73.

[5] This is actually one of the challenges for you when writing your thesis about this topic.

Please follow me on a short sightseeing tour through my collection.

1.1.2 Some Tiny Treasures From My Collection

Do it with matches

The first object I want to present you is more or less a *warm-up*. I owe it to my friend and chess partner Karl Segal (1919–1978). It is one exemplar of a plenitude of puzzles dealing with matches, checkers, coins etc. The famous *Moscow Puzzles* collected by Boris Kordemsky (1972) have a whole chapter devoted to this class of problems, though this one does not appear there.

One evening in 1972 we were sitting in a Viennese chess café and played chess. After an exciting game which I terribly lost Karl wanted to cheer me up. He took a napkin and a pencil and drew an equality sign, Then he grabbed in his pockets and finally found a box of matches, took out a few and arranged them on the napkin as shown in Figure 1.1.

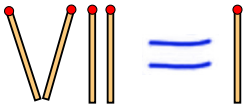


Figure 1.1: A puzzle with matches

Karl explained:

> *Of course, you know how to handle roman numerals. See, this is a mathematical statement, but obviously it is false as* $7 \neq 1$. *You are allowed to move one match, and only one match, so as to make this a true statement. The equality sign must not be touched. So, don't change it into* $\neq$ *by moving a match. Can you solve it? It is not too difficult.*

I have tried to find out the puzzle's origin, but did not succeed. It has been created sometime by some anonymous. Please give it a try. One hint: this puzzle shares a wonderful property with many other puzzles: *The solution lies at an unexpected place.*

Cutting a plate of gold.

Fallacies belong to the basic repertoire of recreational mathematics. They come in various forms: arithmetical fallacies, logical or geometrical fallacies. The latter are particularly interesting and often hidden in puzzles which belong to the class of *dissection problems*.

In practically all books on recreational mathematics you can find the following puzzle, so it's very likely that you have seen it before. Still, I included it into

my exhibition because there's much more behind than a geometrical fallacy. The origin of the puzzle is obscure, but David Singmaster (2014) has found indications that the puzzle may be due to *Sebastiano Serlio (1475–1554)*, an Italian Renaissance architect. In Serlio's 1545 book *Libro Primo d'Architettura* there occurs a geometrical construction which contains a fallacy similar to our puzzle but it passed unrecognized by Serlio. Graham, Knuth, and Patashnik (2003, p. 293) report that this puzzle was Lewis Carroll's favorite[6].

Here's the puzzle:

> *You have a plate of pure gold. It has the shape of a square solid with side length 8 cm and thickness 1 cm. You also have a special high-precision saw to cut the plate which produces no losses due to cutting. Cut the plate as shown in the figure below (left) and rearrange pieces in the way as shown below (right). You will find that the new rectangle has an area of 65 cm^2, whereas the square before cutting had an area of 64 cm^2. So you won one cubic centimeter of gold. Can you explain this miracle?*

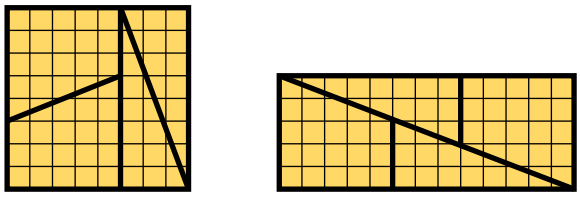
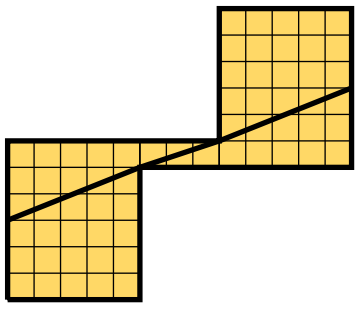


Figure 1.2: Cutting a plate of gold

At the day of writing this (June 2017) the gain is about 700 €. What a fine business idea, money out of nothing!

Can you find an explanantion? No? Not yet? Then you will be surprised that the pieces can be arranged also in this way: Please check, now you have *lost*



one cubic centimeter of gold! But matters are even more mysterious. If you cut a square of size 13×13 in the same way as before (see Figure 1.3), then

[6]Lewis Carroll, pen name of Charles Lutwidge Dodgson (1832-1898), British author (*Alice's Adventures in Wonderland*), mathematician, logician and photographer.

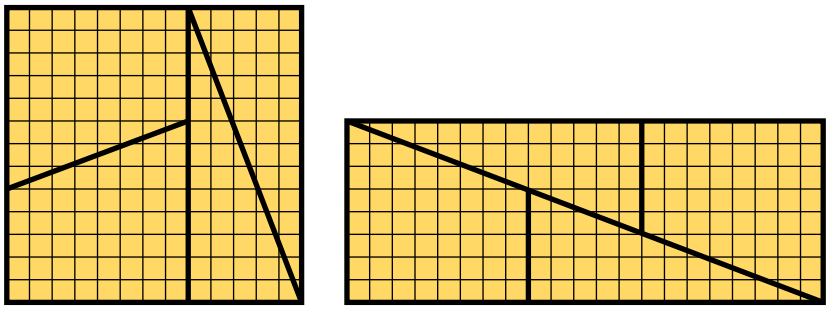


Figure 1.3: Cutting a 13×13 square

you find that after rearranging pieces the resulting rectangle has an area of $168 < 13^2 = 169$. Looking closer at Figure 1.2, you may realize that the pieces in the square have side length 3, 5 and 8. In Figure 1.3, these are 5, 8 and 13. These numbers are members of one of the most famous integer sequences:

$$1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34, \ldots$$

There's an ingenious device available on the internet, *Sloane's On-Line Encyclopedia of Integer Sequences* (`https://oeis.org/`). Just type in the first 5 numbers to learn more about that sequence. You may also consult the bible of discrete mathematics, *Concrete Mathematics* by Graham, Knuth, and Patashnik (2003) where this dissection problem is discussed on page 292.

In this context it is quite illuminating to see a short note by Oskar Schlömilch (1868)[7]. After having described this puzzle to readers of the *Zeitschrift für Mathematik und Physik* he concludes: *Wir theilen diese kleine Leckerei mit, weil die Aufsuchung des begangenen Fehlers eine hübsche Schüleraufgabe bildet und weil sich an die Vermeidung des Fehlers die Lösung und Construction einer quadratischen Gleichung knüpfen lässt.*

So, after all, it is certainly impossible to cut a plane figure, rearrange pieces and the resulting figure has larger area. It's a geometrical fallacy.

Not more?

Well, Ian Stewart (2008, pp. 163) points his readers to a really weird mathematical fact, the *Banach-Tarski Paradox* . In 1924 Stefan Banach and Alfred Tarski[8] proved that it is possible to dissect a sphere into finitely many pieces (actually 5 pieces suffice!), which can then be rearranged to make *two* spheres, each the *same volume* as the original. There are no overlaps, no missing bits, the pieces fit together perfectly. It's a mathematical truth, it can be proved. Still, this fact is so counter intuitive that we call it a *paradox*. It originates from our concept of *volume* and the impossibility of defining this concept in a sensible way for really complicated geometrical shapes.

[7] Oskar Schlömilch, 1823–1901, German mathematician.
[8] Stefan Banach, 1892–1945, Alfred Tarski, 1901 - 1933, Polish mathematicians.

Another Paradox

The origin of this treasure is unknown, the puzzle started spreading around the world in the mid 1990's like a wave of influenza. You can find it for instance in Winkler (2004).

> *You have just moved into an old house with a basement and an attic. There are three switches in the basement marked with ON and OFF. One of the three switches is connected to a bulb in the attic. You have to find out which switch is connected to it. You are allowed to play with the switches for as long as you need to, but you may only go up to the attic once to check the bulb and then say which switch is connected to it. This is the only possibility to find out whether the bulb in the attic is lit.*

Paradoxes convey a counter intuitive fact. This puzzle belongs to a class which can be termed: *I think, I must not have heard correctly!* How can one solve the problem with just *one bit of information*, the latter coming from climbing up to the attic? Still, it is possible.

The Returning Hunter

I found this old riddle in a classic text by Martin Gardner (1994). It runs as follows:

A hunter climbs down from his perch, walks one mile due south, turns and walks one mile due east, turns again and walks one mile due north. He finds himself back where he started. There's a bear at his perch, the hunter shoots the bear. What color is the bear?

So, that was easy! Of course, the bear must be white, it must be an ice bear, since the perch is certainly located exactly at the north pole. Because, otherwise the hunter could not have walked the way described.

But this was not the problem. Here it comes:

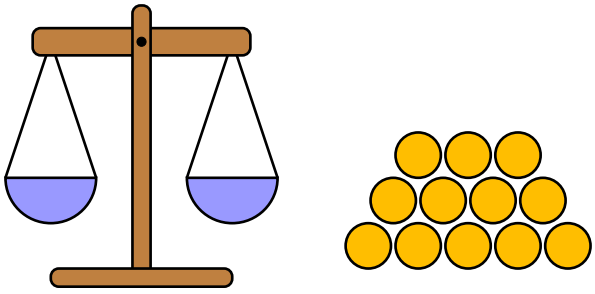
Can you think of another spot on the surface of the earth (assuming it is a perfect sphere), so that you can walk one mile due south, turn and walk one mile due east, turn again, walk one mile due north and arrive at the point where you have started?

If there is such another spot, where is it located? Is there more than one?

The Problem of the Pennies

Let us conclude our sightseeing tour with a real highlight. Our last puzzle is a very prominent one, several famous mathematicians and physicists have worked out solution methods and published papers about it. Still, it comes along in a charming and innocuous manner:

There is a set of 12 pennies, one of them is a counterfeit penny, it may be too light or too heavy. Identify this penny with three weighings only using a simple balance beam. Note, it is unknown to you whether the counterfeit is too light or too heavy.



A variant of this puzzle appears as an unsolved problem posed by Guy and Nowakowski (1995) in the *Problems and Solutions Section* of the *American Mathematical Monthly*. The authors report that this puzzle was very popular on both sides of the Atlantic during World War II. It was even suggested that it should be printed on paper strips and dropped over Germany in an attempt to sabotage their war effort. After the war a series of papers proposing solutions appeared, the most elegant one is due to Freeman J. Dyson (1946), a famous British mathematician and physicist. Besides giving a complete solution Dyson shows that if M equals the number of pennies and there is a number n satisfying

$$M = \frac{1}{2}(3^n - 3), \qquad n = 2, 3, \ldots,$$

then n weighings are always sufficient to identify the counterfeit penny *and* its type. This is exactly the case in our puzzle, for $M = 12$ we have $n = 3$. Dyson also gives a solution for the case $3 \leq M < (3^n - 3)/2$ and shows that the puzzle has no solution if the number of pennies is too large, i.e. if $M > (3^n - 3)/2$.

So much about theory. Now take a sheet of paper and a pencil and try to solve this challenging puzzle. I'm sure you will not prove immune to the fascination of this problem. Have fun!

1.2 Where to go from here

Suppose you have been invited by some renowned magazine to write an article about recreational mathematics. You have been selected for this prestigious job because the editors know about your profound knowledge, your expertise in this field. Thus, the creative challenge is to write a suspenseful report about recreational mathematics!

Here are some suggestions I find interesting and which you may consider. However, you should not feel obliged, your own ideas are certainly welcome.

In any case: please illustrate your work with well-chosen examples and take care to present also elegant solutions, as far as solutions exist. Not every puzzle is solvable, however.

1. **Recreational Mathematics in Education and Teaching**.
   It is a remarkable fact that puzzles and riddles have been used in mathematical education since ancient times. The oldest known example is the *Papyrus Rhind* or *Ahmes Papyrus* which dates around 1550 BC. It contains about 85 exercises in geometry, arithmetics and algebra. Some of them could be called rightly puzzles.

   I think it would be quite challenging to write your thesis about the rôle recreational mathematics plays in education. Puzzles and games are a wonderful way to let pupils participate and at the same time expand their capabilities in problem solving. Bonnie Averbach and Orin Chein have used recreational mathematics in their math lessons over several years. They gathered a lot of experiences which they finally compiled in a remarkable book (Averbach and Chein, 1980). Their teaching paradigm may be condensed in a few motivating sentences addressed to their students: *You participate and be the mathematician. Take a problem and use anything you know to solve it. Think about it; strain your mind and imagination; put it aside, if necessary; keep it in mind; come back to it. If you can solve it on your own, isn't the feeling great? If you can't solve it, maybe some mathematics (new to you) would be helpful to know. let's develop some and see.* It's also my opinion that pedagogically there is really much to be gained from the inclusion of recreational mathematics into your lessons[9].

2. **Famous mathematicians and their puzzles.**
   Famous mathematicians from the days of antiquity up to our time have always taken interest in mathematical puzzles. To name just a few: Archimedes of Syracuse, Cardano, Pascal, Huygens, Newton, Euler, Gauss, Hamilton, Cayley, Sylvester. In the 20th century and in our days: von Neumann, Banach, Littlewood, Ramanujan, Conway, Erdős, Knuth and the physics Nobel-Prize laureate Paul Dirac.

   What are the puzzles and games, they created, on what occasion? In this context the excellent book of Petkovic (2010) will be very helpful.

3. **Famous puzzles and their history.**
   Recall that a good puzzle should have a *raison d'être*. Some of those puzzles gained the distinctive character of being exceptional challenges. They are so interesting that even today many people are discussing them, new

[9] I have three sons, now adult. One became a chemist, one a physicist and one an artist, each of them has a sound mathematical education. During their childhood and youth I regularly entertained them with exciting puzzles, and we really enjoyed. Recently at a family meeting on occasion of an anniversary we remembered those good old days and all those puzzles and riddles. But then one of my sons said: "Dear dad, frankly speaking, sometimes this puzzle stuff was really a torture." Well, perhaps one should not exaggerate it.

solutions are published, variations invented. The *Problem of the Pennies* presented above is a typical representative. *Archimdes' Cattle Problem* is a computationally hard puzzle from number theory with remarkable history. Originally due to Archimedes of Syracuse (287–212 BC), it was rediscovered by Gotthold Ephraim Lessing in 1773 and not solved before 1880. It is still discussed in various mathematical journals. You will have no problem to find more exciting examples. For helpful references please see the annotated bibliography below and search the web.

4. **Classes of puzzles**.
Another way to organize your thesis is to concentrate on a particular *class of puzzles*. There are many such classes: puzzles from number theory, geometry, probability, packing problems, magic squares, paper folding, topology (e.g. knots, Borromean rings), combinatorics and logic (there's some overlap between these classes). Paradoxes and fallacies are another way to categorize puzzles. They have always been very popular. Rich sources of paradoxa and fallacies are geometry, arithmetics, topology and, last but not least, probability theory. One word on topology. In abbreviated form, it deals with properties of geometric forms and space which are invariant with respect to continuous transformations. Today it is a major branch of mathematics, but, interestingly, it has its origin in a mathematical puzzle, *The Seven Bridges of Königsberg*, which has been solved by Leonhard Euler in 1736[10]. Puzzles with a topological background abound, here is one I like very much:

> *Can a tire tube be turned inside out through a hole in its side?*

Hint: Think big!

Or what about *river crossing puzzles* whose origin may be traced back to medieval times, or *train shunting*? Never heard about this? *Train shunting (switching)* problems are particularly popular in Great Britain where they have a very loyal fan base. There are board games, clubs and even regular meetings and conventions devoted to train shunting. The background is a serious one coming from operations research and the good old days of infancy of railroading when there were no double tracks, no automatic switches and no turn tables. Here I have an example from my collection, it comes in many variations (see Figure 1.4 below):

> *At the end of the main line there is a circular track, furthermore there are two wagons and an engine. The objective is to use the engine to change the position of the two wagons and for the engine then to return to the main line. Unfortunately, there is also a low bridge which the engine can pass under, but neither of the two wagons can. The engine can push and pull wagons.*

[10] There will be a topic in this collection on *The Seven Bridges of Königsberg* in near future.

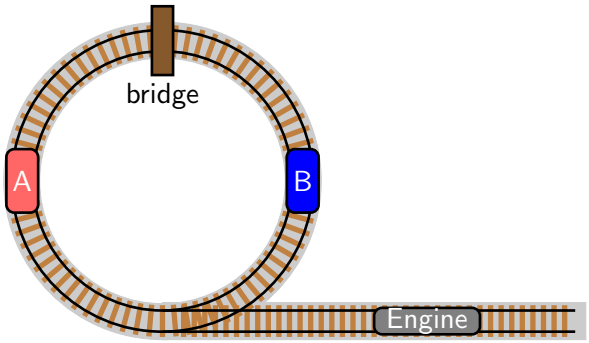


Figure 1.4: Exchange railway wagons A and B

5. Puzzle Composers and Authors.
When reading books or papers on recreational mathematics ever and ever again you will come across a couple of famous names: Henry E. Dudeney, Martin Gardner and Ian Stewart have been mention already. But there is also Sam Loyd, the great American author on puzzle, a somewhat controversial person, a short biography can be found in Gardner (1959b). We should not forget about Lewis Carroll, Peter Winkler, W. W. Rouse Ball and H. S. M. Coxeter, or John Horton Conway. The wide class of logic puzzles is intimately connected with the name of Raymond Smullyan, the great philosopher, logician and piano player. You may not forget about the important and original contributions of French authors. Among these Claude Gaspard Bachet de Méziriac (1581–1638), jesuit clergyman and mathematician, who authored an influential book on recreational mathematics in 1612, *Problémes Plaisants*, which contains e.g., several variants of medieval *river crossing problems*, weighing puzzles and magic squares. Or Jacques Ozanam (1640–1718) who has also a fine book on mathematical puzzles. And, finally I must mention another French author, Edouard Lucas (1842–1898). He has written a classical four-volume book on recreational mathematics and is the inventor of the famous solitaire game *The Towers of Hanoi*.

It would be a fine and promissing idea to develop your thesis around the lives and works of these recognizable people. What was their mathematical background? How did they become authors of texts on recreational mathematics? What about their scientific work apart from recreational maths? What were their greatest successes? Did they have teaching positions? Are there puzzles and games which are inseparably linked with these people?

6. Recreational Mathematics in Media.
As I already remarked in the *Invitation*, newspapers and magazines have always been sources for new and sometimes interesting mathematical puzzles. This has a surprisingly long tradition. A wonderful example in many respects worth to be mentioned is the *Ladies' Diary* or *Woman's Al-*

manack which was published from 1704 until 1840 when it was succeeded by *The Lady's and Gentleman's Diary*. As an almanach it contained calendar information, medical advice, short stories and mathematical puzzles. These were either invented by John Tipper (before 1680–1713), the first editor of the *Diary*, or sent by readers. In issue 25 from 1709 Tipper wrote regarding puzzles: *Arithmetical Questions are as entertaining and delightful as any other Subject whatever, they are no other than Enigmas to be solved by Numbers* (Albree and Brown, 2009). At the beginning puzzles were rather easy but later the level of difficulty increased substantially. Today many periodicals still have puzzle corners, but their character changed somewhat. Now, it's *Sudoku* and its companions like *Kenken, Hashiwokakero* etc. which are dominating the field. I dare not say that these are puzzles in the sense to be discussed in your thesis. They are *tasks*, used mainly to kill time when waiting at airports, for instance. Their availability is almost unlimited because the can be created automatically by diverse computer programs. So, mathematically, they are not very interesting. However, sophisticated puzzles and games continue to be invented and published, a major rôle now playing the world wide web.

I think to examine the issue of perception of recreational mathematics in diverse media today and in the past would be another interesting approach to our topic.

7. And What About Games?

All the puzzles considered so far challenged our ability to reason, it was the delightful play with problems and ideas. Yet, there is another important class of problems in recreational mathematics, *games*. There are *solitaire games*, like Lucas' *Towers of Hanoi*, solitaire with pegs (in may variants), with *polyominoes* (an appealing generalization of dominoes with a surprisingly rich mathematical theory behind), etc. And then, we have games for two players. Now a new element comes into play (literally): one has to account for the ability of reasoning of an *opponent*. In many games the player when it is his turn has a choice between two or more possible moves. Which should be selected? This raises the fascinating problem of finding a *winning strategy*. For some sufficiently simple classes of games it can be proved that such strategies do exist, for others not.

Why not dedicate your thesis to this aspect of recreational mathematics? In almost all books on puzzles you will also find discussions of various sorts of games, see the Annotated Bibliography below. If you decide to pursue this approach then you should have a look at the classical book of Berlekamp, Conway, and Guy (2001–2004).

1.3 An annotated bibliography

You will find that the list of references at the end of this *Topic* is a rather long one. Do not be afraid, there's no need to read all these books and journal articles, it's just an offer. Please read whatever you need for your thesis and whatever you find interesting (maybe all?) and make your choice. Of course, you are free to use other texts and resources.

Let's begin with the *good old books* on recreational mathematics. Edouard Lucas has written classical and often cited textbooks on recreational mathematics, in particular *Récréations mathématiques* (4 volumes, 1882–1894) and *L'arithmétique amusante*, 1895. In volume 3 of his *Récréations* you will find one of the most popular games/puzzles, the *Towers of Hanoi*. Unfortunately, no English translations are available, to the best of my knowledge, but digitized version of the French original texts can be read via the web. One of the grand seigneurs of puzzle and game literature is Henry E. Dudeney. Probably his best-known book is *The Canterbury Puzzles* (1908). The puzzles are presented by characters based from *The Canterbury Tales* by Geoffrey Chaucer (1343-1400), the greatest English poet of the Middle Ages. On-line editions of the collection are available via `www.gutenberg.org`.

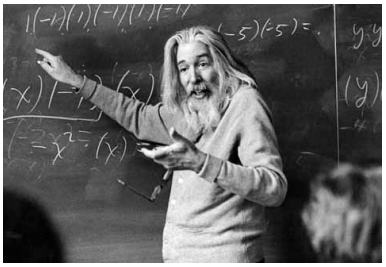
Rouse Ball and Coxeter (1987) is one of those fine books which offer an excellent mix of interesting puzzles and games, partly revivals from very old sources, and nontrivial mathematical theory required to understand and solve the problems presented. My first book on puzzles and games was *The Moscow Puzzles* by Kordemsky (1972). It is still abailable today and offers a wealth of problems and games (and solutions) in 14 chapters. By the way, the English translation has been edited by Martin Gardner.

The book of Averbach and Chein (1980) is written in the same spirit, but the presentation is organized differently. The author's intention is not only to raise interest of students in recreational mathematics, but it also introduces basic concepts needed to solve puzzles. You find there readable short introductions into logic, graph theory, number theory, etc. Thus the emphasis lies on problem solving. Regarding the latter, which as we know is central to mathematical puzzles, you should also read the famous booklet Pólya (2004). It's title is *How to Solve It* and it presents in a charming way rather general guidelines when it comes to solving a mathematical problem. You will enjoy this book.

Martin Gardner's list of publications in recreational maths is quite long. He has been editor of puzzle columns in diverse magazines like *The Scientific American*, many of the problems presented there were later collected in books. Let me mention only a few: *Best Mathematical Puzzles of Sam Loyd* (1959a), *Mathematical Puzzles and Diversions*, (1959b), *Hexaflexagons, Probability Paradoxes, and the Tower of Hanoi*, (2008).

Regarding puzzles and games from logic I recommend reading the excellent and entertaing books by Raymond M. Smullyan. He has authored more than 30 books about logic and logic puzzles. At the first place I should mention the

rather recent *Gödelian Puzzle Book* (2013). Here you can find entertaining variations of Kurt Gödel's *Incompleteness Theorems*, puzzles related to basic concepts of modern logic like truth, provability and undicidability. You may find amusing and interesting also *What is the Name of This Book?* (2011) and *The Lady or the Tiger?* (1982). His many puzzles about *truth-tellers and liars* have become really famous. Some of his books are available in the web.



Raymond M. Smullyan
1919-2017

Fine collections of rather recently invented puzzles are the books by Peter Winkler (2004) and (2007). Winkler is professor of discrete mathematics at the Dartmouth College. Berlekamp, Conway, and Guy (2001–2004) is *the* textbook on mathematical games. Volumes 1-3 deal with strategies for two-person games, volume 4 is devoted to solitaire games, topological puzzles and even *Rubik's Cube* finds its place there. Last but not least I want to point you to Petkovic (2010). This book is a collection of stories and puzzles due to great mathematicians. Carefully worked out solutions are given also.

1.4 References

[1] Joe Albree and H. Brown Scott. "A valuable moment of mathematical genius: The Ladies' Diary (1704-1840)". In: *Historia Mathematica* (2009), pp. 10–47.

[2] Bonnie Averbach and Orin Chein. *Problem Solving Through Recreational Mathematics*. Dover Publications, 1980.

[3] G. P. Basharin, A. N. Langville, and V. A. Naumov. "The Life and Work of A. A. Markov". In: *Linear Algebra and its Applications* 386 (2004), pp. 3–26.

[4] E. R. Berlekamp, J. H. Conway, and Richard K. Guy. *Winning Ways for your Mathematical Plays*. 2nd. 4 vols. A. K. Peters, 2001–2004.

[5] Henry E. Dudeney. *The Canterbury Puzzles*. E. P. Dutton and Company, 1908. URL: http://www.gutenberg.org/files/27635/27635-h/27635-h.htm.

[6] Freeman J. Dyson. "The problem of the pennies". In: *The Mathematical Gazette* 30.291 (1946), pp. 231–234.

[7] Martin Gardner. *Best Mathematical Puzzles of Sam Loyd*. Dover Publication, 1959.

[8] Martin Gardner. *Hexaflexagons, Probability Paradoxes, and the Tower of Hanoi*. Cambridge Universty Press, 2008.

[9] Martin Gardner. *Mathematical Puzzles and Diversions*. Penguin Books, 1959.

[10] Martin Gardner. *My Best Mathematical and Logic Puzzles*. Dover Publications, 1994.

[11] L. Graham Ronald, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. 2nd ed. Addison-Wesly, 2003.

[12] Richard K. Guy and Richard Nowakowski. "Coin-Weighing Problems". In: *American Mathematical Monthly* 102.2 (1995), pp. 164–167.

[13] Boris A. Kordemsky. *The Moscow Puzzles*. Dover Publications, 1972.

[14] Miodrag S. Petkovic. *Famous Puzzles of Great Mathematicians*. American Mathematical Society, 2010.

[15] G. Pólya. *How to Solve It*. Princeton University Press, 2004.

[16] W. W. Rouse Ball and H. S. M. Coxeter. *Mathematical Recreations and Essays*. 13th. Dover Publications, 1987.

[17] O. Schlömilch. "Ein geometrisches Paradox". In: *Zeitschrift für Mathematik und Physik* 13 (1868), p. 162.

[18] David Singmaster. *Vanishing Area Puzzles*. 2014. URL: http://rmm.ludus-opuscula.org/Home/ArticleDetails/94.

[19] Raymond M. Smullyan. *The Gödelian Puzzle Book*. Dover Publications, 2013.

[20] Raymond M. Smullyan. *The Lady or the Tiger?* Dover Publications, 1982.

[21] Raymond M. Smullyan. *What is the Name of This Book?* Dover Recreational Math, 2011. URL: https://archive.org/details/WhatIsTheNameOfThisBook.

[22] Ian Stewart. *Professor Stewart's Cabinet of Mathematical Curiosities*. Perseus Books, 2008.

[23] Peter Winkler. *Mathematical Mind-benders*. Taylor and Francis, 2007.

[24] Peter Winkler. *Mathematical Puzzles: A Connoisseur's Collection*. A. K. Peters, 2004.

[25] Peter Winkler. "Review of *Famous Puzzles of Great Mathematicians* by Miodrag S. Petkovic". In: *The American Mathematical Monthly* 118.7 (2011), pp. 661–664.

Topic 2

Shortest Paths in Networks

It's one thing to feel that you are on the right path, but it's another to think yours is the only path.

Paulo Coelho, 2006

Keywords: *combinatorial optimization, graph theory, algorithms computer science*

2.1 An Invitation

2.1.1 The problem and its history

One of my sons is living with his family in a small village close zu Zürich. From time to time I set out to visit him and normally, I travel by airplane because it takes only one hour flight time. But I could also use the car. In this case I have to expect eight or even more stressful hours of driving. On the rare occasions when I decide to use the car, then, of course, I want to drive a shortest route, so I use a navigation system. Typing in the point of departure and the destination of my journey, the navigation system calculates two suggestions of optimal routes I could drive, and this takes only fractions of a second.

How can this be done so quickly?

A naïve approach to carry out calculations is this one:

- Find *all* possible routes and calculate their lengths.
- Select the route with shortest length.

This idea is not a good one: even for moderate sized networks of roads the number of possible routes is in general an extremely large number. Furthermore, most of these routes are not worth to be considered. For instance, travelling from Vienna to Zürich via Milan hardly makes any sense.

Modern navigation systems do their job in a completly different way, they use an *algorithm* for solving a *shortest path problem* (SPP).

Any famous problem has its history. But for the SSP it is rather difficult to pin down the origins of the problem exactly because we have almost no written

evidence from ancient times. There is an exception, though, the SPP does occur *implicitly* in a few medieval texts, sometimes hidden in some sort of *puzzle*.

This is of course a curiosity. But seriously, we may imagine that even in very primitive societies finding shortest paths was an essential task for gathering food, distributing goods, i.e., trading, and for communication. Besides human beings also animal societies are confronted with SPPs. Due to evolution certain animal societies are really high performers when solving SPPs. An interesting and striking example is the argentine ant, *Linepithema humile*. These ants form mega colonies of monstrous size. One of these mega colonies ranges over more than 6000 km from the northern parts of Spain to the south of Italy and its subcolonies are connected by well organized and optimized paths. Thus it should not come as a surprise that in modern combinatorial optimization *metaheuristics* based on ant colonies are routinely applied to solve very difficult and large scaled vehicle routing problems (which involve SSPs, of course).

The modern theory of shortest paths has its origins in the 1950's when providers of communication systems were facing an enormous growth of traffic volume. For instance, when at that time a customer made a long-distance call the major difficulty was to get the call to its destination. If the route of first choice was busy then the operators had to send the call along a second best route, a third best, etc. In the process of automation of communication it was necessary to program telefone exchange facilities to find such alternate routes quickly. This certainly involves solving nontrivial SPPs. So, it is not surprising that within a couple of years many people came up independently with almost the same algorithms for solving SPPs. An interesting account of the developments during these years is Schrijver (2012).

Today SPPs are among the most important and most fundamental optimization problems. They are intesting and challenging not only *per se*, but occur ever and ever as subproblems in more complex settings. The latter range from vehicle routing and related transportation problems, the analysis of large social networks, molecular biology (DNA sequence alignment) to the development of highly integrated microprocessors. Since the networks arising in these applications are usually of extraordinary size, efficiency of algorithms is certainly a major issue.

2.1.2 Preparing the stage - graphs, paths and cycles

The mathematical structure underlying the SPP is that of a *graph*. In this subsection I'll introduce to you a few important definitions and basic terminology. As the focus of this *Invitation* does not lie on mathematical rigor we may approach the concept of a *graph* in a rather pedestrian-like fashion.

A *graph* is a set of points V where some pairs of these points are *related* in a certain way. A most natural way to visualize this concept is a *roadmap*: the points correspond to villages of junctions of roads in a certain area. Two points are related, if there exists a road connection between these points. This idea of a network of roads appears as early as 1736 when Leonhard Euler (1707-1783)

solved the famous problem of the *Seven Bridges of Königsberg*. Indeed, Euler's work marks the beginning of a mathematical discipline known today as *Graph Theory*[1].

The points in the set V will be called *vertices*. If two points (locations) $u, v \in V$ are related, e.g. by a road connection, then we call the *ordered pair* (u, v) an *arc* and denote the set of all arcs by A. Note carefully, that these pairs (u, v) are considered to be ordered, i.e., $(u, v) \neq (v, u)$. Therefore, in the context of a network of roads arcs represent *one-way roads*.

More formally, a *graph* G is an ordered pair $G = [V, A]$ of the set of vertices and the set of arcs. The number of vertices $n = |V|$ will be called the *order* of the graph G, the number of arcs $m = |A|$ the *size* of G.

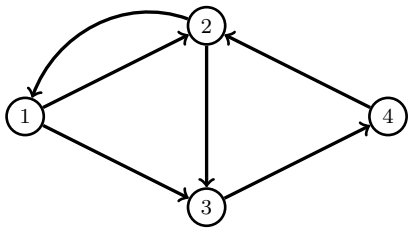
As long as the order n of G is not too large we may draw a diagram of G. This is simply done in the following way:

- Draw n labeled points in the plane, these are the vertices.
- For each arc $(u, v) \in A$ draw an arrow connecting vertices u and v such that the arrow points from u to v.

Here is an example: let $V = \{1, 2, 3, 4\}$ and define the set of arcs by

$$A = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 4), (4, 2)\}$$

Then one possible diagram of $G = [V, A]$ would be:



A little bit more has to be said about the arcs of a graph.

- Our definition allows A to equal the empty set $\varnothing$, but normally A will be nonempty and contain $m = |A| > 0$ arcs.
- If there exists a pair $a = (u, v)$ in A then we say that the arc a connects the vertices u and v. It is also customary to say that a is *incident from* u and *incident to* vertex v. For instance in our example above there is an arc $a = (2, 3)$, thus a is incident from vertex 2 and incident to vertex 3.
- We assume that an arc never connects a vertex with itself, so there are no *loops* (u, u).
- Furthermore, as A is a set, no arc occurs more than once in A. It is possible to extend the definition of a graph so that its arc set A becomes a *multiset* with multiple occurrences of arcs but we will not do so in this *Invitation*[2].

[1] There will be a *Topic* devoted to the Königsberg Problem.

[2] Such parallel arcs are interesting in certain routing problems, e.g., the *Chinese Postman Problem*.

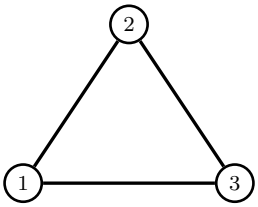
If for each arc $(u, v) \in A$ there is also its opposite $(v, u) \in A$, then G is called an *undirected graph*, otherwise G is a *directed graph*, sometimes called a *digraph*. It helps intuition to think of a directed graph as a system of oneway roads, whereas an undirected graph may be seen as a network of roads each being twoway. There is a one-to-one correspondence between directed und undirected graphs: if we add to A for each arc (u, v) its opposite (v, u) unless it is already in A, then we obtain an undirected graph. This is a pretty simple idea but unfortunately, it has its limitations in a shortest path context.

For definiteness: in all what follows, when the term *graph* occurs, then it alway means directed graph, unless stated otherwise.

Here is an example of an undirected graph: $V = \{1, 2, 3\}$, and arc set

$$A = \{(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2)\}$$

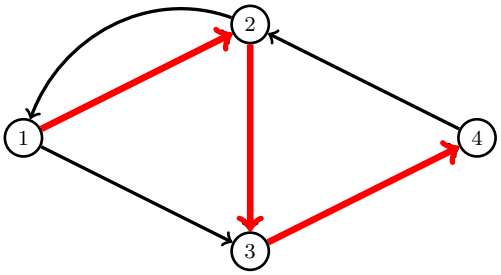
When drawing a diagram of an undirected graph it is customary to draw a single line segment without arrow tips of each pair of arcs $\{(u, v), (v, u)\}$:



Let $G = [V, A]$ be a graph. A *path* P from vertex s to some vertex t is a sequence of contiguous arcs

$$P = [(s, a), (a, b), (b, c), \ldots, (y, z), (z, t)]$$

P is a *simple* path if does not use the same arc more than once, P is *elementary*, if it does not use the same vertex more than once.



A path P from $1 \to 4$

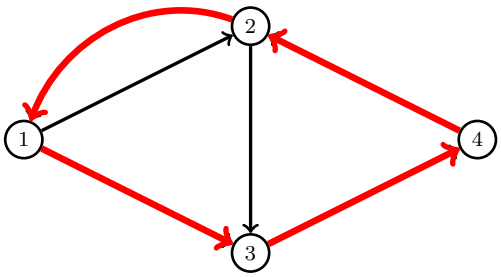
$P = [(1, 2), (2, 3), (3, 4)]$

P is simple and elementary

The path P in the graph displayed above may be written more compactly as $P = [1, 2, 3, 4]$.

If for a path P initial vertex and terminal vertex coincide, then P is called a *cycle*. For instance $C = [1, 3, 4, 2, 1]$ is a cycle[3]:

[3] **A note on terminology.** Many authors prefer the term *circuit*, but we use the term



Note that in this graph $[1, 2, 1]$ is also a cycle, a *2-cycle*.

A graph is *weakly connected*, if for each pair of vertices $u, v \in V$ there exists *either* a path from $u \to u$ *or* from $v \to u$. A graph is called *strongly connected*, if for each pair of vertices there exists a path from $u \to v$ *and also* a path from $v \to u$. It easy to verify by inspection that the sample graph given above is strongly connected. But in general, proving connectedness is a rather nontrivial task. In the sequel we will always assume that the graphs we are dealing with are at least weakly connected.

2.1.3 Weighted graphs

Let $G = [V, A]$ be a graph and assign *weights* to arcs. Each arc $(u, v) \in A$ is attached a weight $w(u, v)$ which we assume to be an *arbitrary* real number. Let me now present some examples, and I will use this opportunity to show you which meaning we can give the notion of *length* of a path.

Example 1. (Transportation)

In a transportation network the weights most often represent physical distances between destinations. But weights may also be cost of transportation, as it is the case with the graph in Figure 1.1. The weight $w(3, 6) = 2$ means that transportation from vertex 3 to vertex 6 costs 2 € per unit, whereas when transporting goods from 3 to 2 we make a profit of 4 € per unit because $w(3, 2) = -4$.

In this example it is most natural to define the length $\ell(P)$ of a path P as the sum of its constituent arcs. Thus for a path $P = [v_0, v_1, \ldots, v_k]$ we define:

$$\ell(P) = \sum_{i=1} w(v_{i-1}, v_i).$$

For instance, the path $P = [1, 3, 2, 5, 6]$ in the graph displayed in Figure 1.1 has

cycle, as it prevails in the literature on shortest paths. Interestingly, although the theory of graphs is a rather mature field of mathematics there is still some babylonian confusion. Indeed, Richard Stanley (MIT and Clay Institute) once said: *The number of systems of terminology presently used in graph theory is equal, to a close approximation, to the number of graph theorists.*

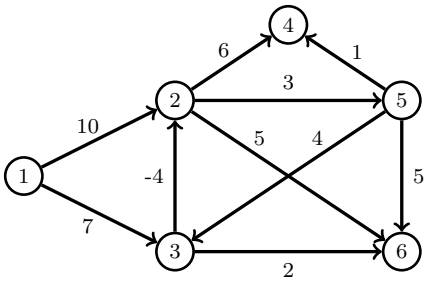


Figure 2.1: A transportation network

length

$$\ell(P) = 7 + (-4) + 3 + 5 = 11.$$

By simple inspection you will find that this P is not the *shortest path* from $1 \to 6$.

Example 2. (Reliability)

The weights assigned to arcs of a graph may also be *probabilities*. When does this make sense?

Consider for instance a communication network. We may model this as a graph with vertices representing transmitters or relay stations, arcs are radio or cable connections. An interesting type of weight of the arc (u, v) is the maximum capacity of the wire connecting vertices u and v. But here we are more interested in the *reliability* of a connection, the probability that the connection is available at a particular time.

Figure 1.2 gives an example of a small communication network.

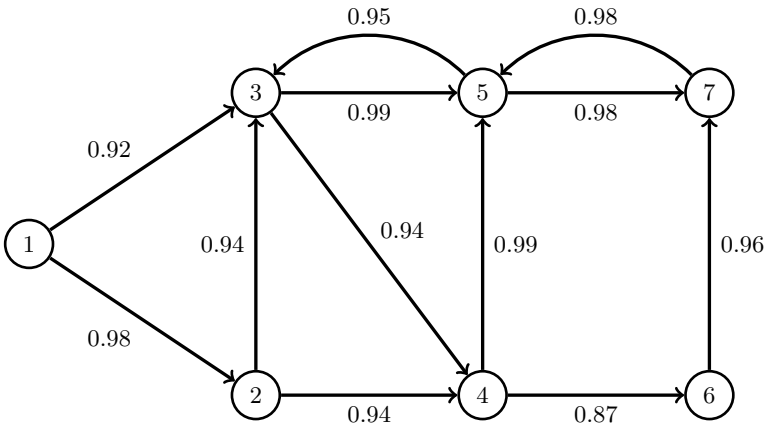


Figure 2.2: A communication network

Consider the path $P = [1, 2, 3, 5, 7]$ in the corresponding graph. What is the reliability of this path? Assuming statistical independence, we would calculate:

$$R(P) = 0.98 \cdot 0.94 \cdot 0.99 \cdot 0.98 = 0.89375$$

In other words, the reliability of a path is just the *product* of the weights of those arcs which lie on the path P. Of course, we would be interested in finding a path of *maximum reliability*.

The problem now looks somewhat different from that in Example 1. But this is not so. We may most easily transform our problem to that of finding a shortest path. Just take logs!

If we define the reliability of $P = [v_0, v_1, \ldots, v_k]$ as

$$R(P) = \prod_{i=1}^{k} w(v_{i-1}, v_i),$$

then

$$\ln R(P) = \sum_{i=1}^{k} \ln w(v_{i-1}, i)$$

There are two important observations you can make at this point:

- Maximizing $R(P)$ is equivalent to maximizing $\ln R(P)$, since the logarithm is a monotone increasing function.
- Because arc weights $w(u, v)$ are probabilities, we have $0 < w(u, v) \leq 1$. But this implies that the logs are negative: $\ln w(u, v) \leq 0$ for all arcs in the graph.

Therefore

$$R(P) \to \max \quad \Leftrightarrow \quad \ln R(P) \to \max \quad \Leftrightarrow \quad -\ln R(P) \to \min$$

In other words, if we replace the arc weights $w(u, v)$ by $-\ln w(u, v)$ then finding a path of maximum reliability becomes a shortest path problem!

Example 3. (Trading currencies)

This beautiful example is taken from Sedgewick and Wayne (2011, p. 679). Trading currencies is an important branch of business for many big banks like Deutsch Bank and others. The basic idea lying at the heart of this business is to exploit short term deviations from equilibrium state on foreign exchange markets. To see how it works consider a set of five currencies: U.S. dollars (`USD`), Euros (`EUR`), British pounds (`GBP`), Swiss francs (`CHF`) and Canadian dollars (`CAD`). At a particular day[4] the following exchange rates between these currencies were:

[4] Unfortunately the authors do not disclose to us the exact date when these data have been recorded.

	USD	EUR	GBP	CHF	CAD
USD	1	0.741	0.657	1.061	1.005
EUR	1.349	1	0.888	1.433	1.366
GBP	1.521	1.126	1	1.614	1.538
CHF	0.942	0.698	0.619	1	0.953
CAD	0.995	0.732	0.650	1.049	1

Suppose now, we want to change 1000 USD into CAD. How much Canadian dollars do we get? Easy, just calculate:

$$1000 \text{ USD} = 1000 \cdot 1.005 = 1005 \text{ CAD}$$

Can we get more? Let's try this: first convert USD into Swiss francs and then we convert these into Canadian dollars:

$$1000 \text{ USD} = 1000 \cdot 1.061 \cdot 0.953 = 1011.1 \text{ CAD}$$

You see: it makes a differences. Alternatively, we may convert to EUR first and then to CAD:

$$1000 \text{ USD} = 1000 \cdot 0.741 \cdot 1.366 = 1012.2 \text{ CAD}$$

This is again a little bit more, compared to direct conversion USD $\to$ CAD we have a plus of 0.72 percent. Not very much it seems, but thinking of a global player moving around billions of dollars, the situation appears in another light.

Now, it is quite natural to ask: *is there an optimal strategy of converting currencies*? We do not want to rely on trial and error any longer, we need a mathematical model.

Let's model the currency exchange problem by a graph: as vertices we take the five currencies. Each vertex is connected to every other vertex by an arc, because we can convert every currency in any other. The weights $w(u, v)$ of the arcs are the exchange rates from currency u to currency v given in the table above. Figure 1.3 shows a diagram of this graph. Note that I have drawn this graph in such a way that the orientations of the arcs are encoded in the arc labels together with exchange rates. Otherwise the diagram would have become too messy to be useful.

A sequence of conversions corresponds to a path in this graph, the *weight* of the path is the *product* of the exchange rates along the path and equals the total exchange rate along that path. For instance, the path $P = [\text{USD}, \text{EUR}, \text{CAD}]$ has a weight of $w(P) = 0.741 \cdot 1.366 = 1.0122$.

For a path $P = [v_0, v_1, \ldots, v_k]$ the total exchange rate equals:

$$E(P) = \prod_{i=1}^{k} w(v_{i-1}, v_i)$$

Of course, we want paths which maximize $E(P)$. As in Example 2 by taking logarithms of weights and changing their signs we turn this maximum problem

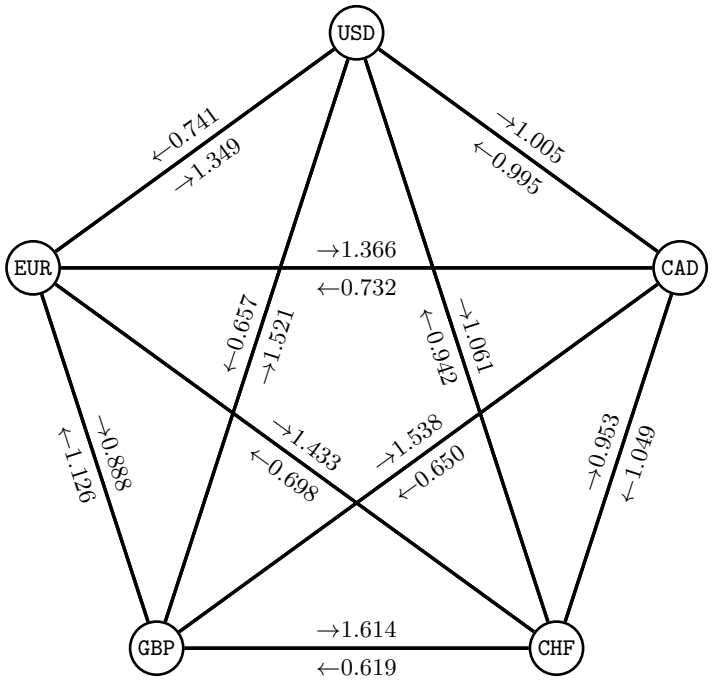


Figure 2.3: The currency exchange problem

into a problem of finding a shortest path:

$$E(P) \to \max \quad \Leftrightarrow -\ln E(P) = -\sum_{i=1}^{k} \ln w(v_{i-1}, v_i) \to \min$$

At first sight the situation looks completly analogous to that encountered in Example 2. But there is a subtle difference. In Example 2 the weight transformation $w(u, v) \mapsto -\ln w(u, v)$ resulted in weights all nonnegative because weights were probabilities. Here this is no longer the case. The transformed weights may be positive or negative. This has serious consequencies.

Example 4. (Scheduling)

This nice example has been adapted from Gondran and Minoux (1995, pp. 65). The construction of a single-family house requires the performance of a number of certain tasks like masonry, making the roof, sanitary installations, etc. These tasks cannot be performed in arbitrary order. For instance the carpentry of the roof requires a greater part of masonry to be finished. The following table gives a (highly aggregated and simplified) list of tasks, their duration in days and their dependencies:

Task Nr.	Task	Duration	Previous tasks
1	masonry	10	–
2	carpentry of roof	3	1
3	tiling of roof	1	2
4	sanitary and electrical installations	8	1
5	front	2	3,4
6	windows	1	3,4
7	garden	4	3,4
8	ceiling	3	6
9	painting	2	8
10	moving in	1	5, 7, 9

Let's represent the project of building a house by constructing a *precedence graph* $G = [V, A]$. This allows us to take explicitly care of dependencies of tasks. G has as vertices the 10 tasks, so $V = \{1, 2, \ldots, 10\}$. Whenever a task v requires another task u to be finished, then this induces an arc $(u, v) \in A$ in G. These arcs we assign weights equal to the duration of task u.

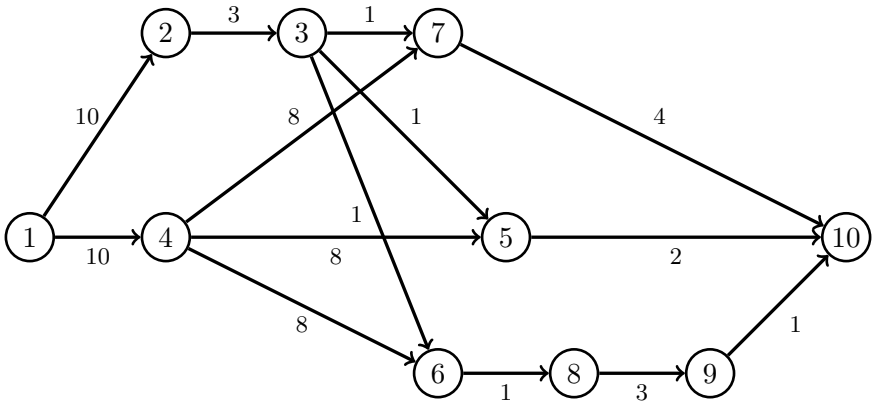


Figure 2.4: The precedence graph for building a house

See Figure 2.4 for a diagram of this precedence graph.

One of the most important properties of precedence graphs is that they cannot have cycles. Looking at Figure 2.5 reveals immediately why this is so.

If we try to interpret the graph in Figure 2.5 as precedence graph we run into serious trouble, for it says:

- Task 2 cannot start before task 1 has been finished.
- Task 3 cannot start before task 2 has been finished.
- Task 1 cannot start before task 3 has been finished ???

The last statenment says that task 1 *precedes itself* which is impossible. Proper precedence graphs are so-called *DAGs*, an acronym for directed acyclic graph.

What about paths and their lengths in a precedence graph?

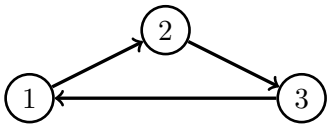


Figure 2.5: Precedence graphs must be acyclic.

Consider for instance the path $P_1 = [1, 4, 5]$ in Figure 2.4, it has total length $\ell(P_1) = 18$. This tells us that performing tasks 1 and 4 in this order will take 18 days. But can we start working on task 5 immediately after task 4 has been completed?

No! From the precedence graph we can read off that task 5 also depends on task 3, this in turn depends on task 2. So we find: The earliest time that task 5 can be started is the *length of the longest path* from 1 to 5 in the graph because this takes care of all activities task 5 is depending on.

This fact is of considerable significance in *project scheduling*: any activity or task u represented by a vertex in a precedence graph cannot be started earlier than the length of some longest path to u. In our example vertex $u = 10$ is of particular interest, because the length of a longest path from $1 \to 10$ gives us the *makespan* of building the house, the earliest time (counted from $t = 0$) that the house is finished.

So, in Example 4 we end up with a *longest path problem*[5].

Is it possible as in Examples 2 and 3 to transform this problem into a shortest path problem? Yes, and that is again very easy, just negate all arc weights, i.e., apply the transformation $w(u, v) \mapsto -w(u, v)$. In general, as the next section shows, this transformation leads into serious trouble. But for precedence graphs it works fine, because precedence graphs have no cycles, they are DAGs.

2.1.4 Solvability

In the last section we have seen that several optimal path problems can be reduced to the SSP. Thus at first sight it seems more or less obvious that the SSP must have a solution for any given weighted graph. But unfortunately, this is not so. The problem arises only when we have *negative weights*. The following small example shows why. What is a shortest path from $1 \to 4$?

Well, guided by intuition you may suggest that $P = [1, 2, 3, 4]$ is a shortest path with length $\ell(P) = 20$. But is it really the shortest path from $1 \to 4$?

No! Consider $Q = [1, 2, 3, 5, 2, 3, 4]$. This path has length $\ell(Q) = 15$, so Q is shorter than P. But Q isn't a shortest path either because the path $R = [1, 2, 3, 5, 2, 3, 5, 2, 3, 4]$ is even shorter, it has length $\ell(R) = 12$.

Now you can see the problem very clearly: the paths R and Q contain the

[5]Eric Denardo once ironically remarked that only a notorious pessimist can be interested in longest paths.

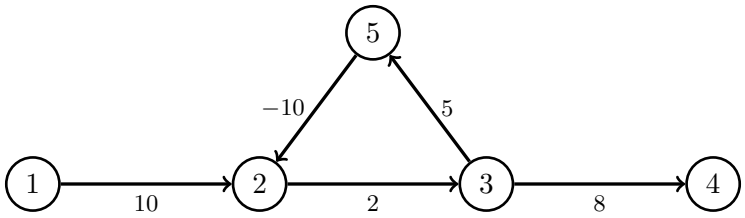


Figure 2.6: There's a problem with this graph!

cycle $C = [2, 3, 5, 2]$ which has *negative* length $\ell(C) = -3$. Each time a path traverses the cycle C the length of a path $1 \to 4$ is reduced by 3 units. Since we can traverse this cycle as often as we like, it follows that the length of any path $1 \to 4$ can be made arbitrarily small. In other words, the SSP has no solution for the graph given in Figure 1.4!

Thus we have the important result: whenever a connected graph contains negative cycles then the SSP has no solution, one also says the SSP is *an ill posed problem*.

Although the SSP has no solution, the existence of negative cycles in a graph is a truly significant message, it tells us something very important about the problem at hand!

To see this, let's look once again at the currency exchange problem discussed in Example 3. Consider the path $P = [\texttt{USD}, \texttt{EUR}, \texttt{CAD}, \texttt{USD}]$ which is a cycle. After transforming the exchange rates $w(uv) \mapsto -\ln w(u, v)$, we find that P has length

$$\ell(P) = -\ln(0.741) - \ln(1.366) - \ln(0.955) = -0.0071196,$$

so this cycle has negative length. What does it mean? At the first place it means that the SSP on this graph has no solution. There exists no (finite!) optimal path to convert `USD` to `CAD`.

At the second place the existence of a negative cycle opens a fascinating economic perspective! The total exchange rate along this cycle equals

$$E(P) = e^{-\ell(P)} = e^{-0.0071196} = 1.0071$$

In other words, starting with 1000 `USD`, converting these to `EUR`, then `EUR` to `CAD` and these back to `USD` we get 1007.1 `USD`. Of course the profit does not appear to be very high. But you may bear in mind that a trader with an initial capital of `USD` 1 000 000 can make a profit of `USD` 1 007.1 every minute, about `USD` 420 000 per hour! In practice the so-called *arbitrage profit* is limited only by the time required to perform the exchanges, but using high frequency trading devices the profit may be extremly large, indeed.

Thus we get *money out of nothing!* What a fine business idea.

Frankly speaking, the picture I have drawn is not a complete one as it does not account for *transaction cost*. But if the trader is a global player like Deutsche

Bank then transaction cost can be kept at minimum level. Thus the arbitrage business is plenty profitable in the real world.

Let us pause for a moment here. In this section we have seen that the SSP has a solution if the underlying graph has no cycles of negative length. Thus an *algorithm* for the SSP should be capable of:

- Detection of negative cycles;
- Efficient determination of a shortest path.

In the next section I will introduce you to a very general idea to cover both issues.

2.1.5 It's time to relax

Let $G = [V, A]$ be a directed and connected graph with given arc weights. Before we embark on formulating an algorithm for the SSP we should become clear about what we *really want*:

- If there is a negative cycle then it should be detected.

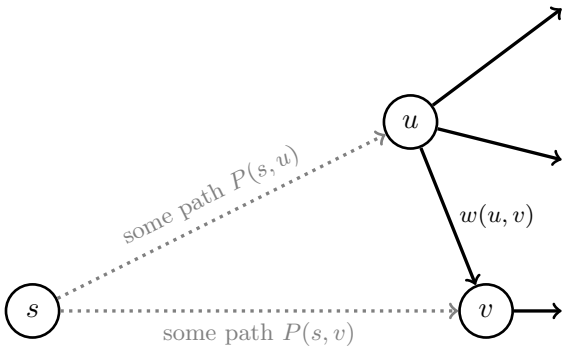
  otherwise

- We may fix two vertices s and t and find the shortest path connecting s and t. This is known as *single-pair shortest path problem* (SPSP).
- We may fix *one vertex* s and want to find the shortest paths *to all other vertices*, commonly refered to as *single-source shortest path problem* (SSSP).
- Alternatively we want to determine the shortest paths between *all pairs* of vertices in G, the *all-pairs shortest path problem* (APSP).

Quite remarkably, from a computational point of view finding shortest paths from one source to all other vertices is not significantly more expensive than solving the single-pair problem. In this section we shall concentrate on SSSP and defer the all-pairs problem to Section 2.

A majority of algorithms for the single-source problem are based on a very simple and effective idea: relaxation of arcs. The origin of this idea is not completely clear. As far as I could find out it appeared for the first time in Ford (1956), and independently in the 1958 French edition of Berge (1962, p. 70).

Relaxation is a very simple and intuitive concept: suppose we have found a path $P(s, u)$ connecting s and u with length $d(u)$ and a path $P(s, v)$ to v having length $d(v)$. Neither $P(s, u)$ nor $P(s, v)$*need be shortest paths*! Suppose further that there exists an arc (u, v) with length $w(u, v)$.



Relaxing the arc (u, v) means that we test whether we can improve the path to v found so far by going through u. This would be the case, if

$$d(v) > d(u) + w(u, v). \tag{2.1}$$

If inequality (2.1) happens to hold then *just take the shortcut* by going to v via u, that's the idea! Easy, isn't it?

Any arc (u, v) satisfying (2.1) is called *eligible* for relaxation, otherwise *ineligible*.

Now let's craft this into an iterative algorithm. This is apart from implementation details (some of which will be discussed shortly) the famous *Bellman-Ford Algorithm*[6].

For our algorithm to run we need two vectors d and p, each of dimension $n = |V|$, the number of vertices.

- The vector $d = [d(1), d(2), \ldots, d(n)]$ holds the lengths of the shortest paths from s to any other vertex found so far. We initialize the distance vector by:

$$\begin{array}{rcll} d(s) & = & 0 & \\ d(k) & = & \infty & \text{for } k \neq s \end{array}$$

- The second vector p is used to keep track of shortcuts. Initially all components of p are set to zero. Whenever an arc (u, v) is relaxed then we set $p(v) = u$ thus indicating that going to v is shorter by passing through u and then take arc (u, v). We will need p to *reconstruct* the shortest paths from s to any other vertex.

After initialization we perform the following two steps:

(A) Find *any* eligible arc (u, v). If there is none, then STOP.
(B) If (u, v) is eligible, then set $d(v) = d(u) + w(u, v)$, update the predecessor vector p by putting $p(v) = u$ and return to **(A)**.

Before we give this algorithm a try, we have to discuss briefly some important questions.

[6] Richard Bellman (1920–1984), Lester Randolph Ford jun. (1927–)

1. Does this algorithm actually find all shortest paths from s to any other vertex in G?

Yes, unless G has a negative cycle because, as we already know, in this case the SPP is ill posed and has no solution. As the algorithm is *iterative*, finding the solution requires the algorithm to *converge* meaning that the sequence of distance vectors has a limit. An informal argument for convergence is this: at any stage of the algorithm the distances in vector d are *bounded from below* by the actual minimum lengths of shortest paths. It can never happen that some $d(u)$ is smaller than the length of a shortest path from $s \to u$. Furthermore, the relaxation condition (2.1) guarantees that the values in vector d will *decrease monotonically*. Note that this is an informal and incomplete argument. Of course, convergence of the Bellman-Ford Algorithm requires a rigorous proof, please see Section 2.

2. In which order?

In step (A) we required to find *any* eligible arc, nothing was said about how and in which order eligible arcs are processed. The Bellman-Ford Algorithm resolves this ambiguity by first forming a list of all arcs in *some order*, there are $m = |A|$ of them, and then processing them one after the other by checking eligibility. By organizing search for eligible arcs in a more sophisticated manner we get other algorithms for SPP which are more efficient than Bellman-Ford. See Section 2.

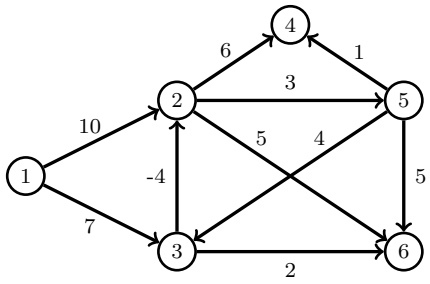
3. When to stop?

We stop when there are no more eligible arcs. But how many iterations are necessary? Suppose that there is no negative cycle, then a shortest path from s to some vertex v cannot have more than $n-1$ arcs. For, if it had, then the path would have passed through at least one vertex more than once. But that can happen only if it passed through a cycle. If there are only cycles of length ≥ 0 then the relaxation (2.1) would have been violated. So that's impossible. On the other hand, if there are negative cycles, we will always find eligible arcs and the algorithm will never stop unless we *force* it to do so. In the Bellman-Ford Algorithm this is simply done by:

- Process the list of all arcs $n-1$ times. Stop earlier when there are no more eligible arcs.
- Check if there is still an eligible arc. If so, then we have encountered a negative cycle.

2.1.6 A sample run of the Bellman-Ford Algorithm

Let's return to the simple transportation network of Example 1 and let's find all shortest paths from vertex 1 to any other vertex. The weighted graph $G = [V, A]$ of this network is repeated here for ease of reading:



The list of arcs and their weights is:

(u,v)	$(1,2)$	$(1,3)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,2)$	$(3,6)$	$(5,3)$	$(5,4)$	$(5,6)$
$w(u,v)$	10	7	6	3	5	-4	2	4	1	5

We initialize the distance and predecessor vectors d and p:

u	1	2	3	4	5	6
d	0	∞	∞	∞	∞	∞
p	0	0	0	0	0	0

Now we perform a *first pass* through the list of arcs. Each of them is eligible for relaxation, so we get step by step:

$$
\begin{aligned}
\text{arc } (1,2): \quad & d(2)=\infty > d_1 + w(1,2) = 0 + 10 = 10 \\
& \qquad \text{thus relax } (1,2) \text{ and put } d(2) = 10, p(2) = 1 \\
\text{arc } (1,3): \quad & d(3)=\infty > d_1 + w(1,3) = 0 + 7 = 7 \\
& \qquad \text{thus relax } (1,3) \text{ and put } d(3) = 7, p(3) = 1 \\
& \dots \\
\text{arc } (3,2): \quad & d(2)=10 > d_3 + w(3,2) = 7 + (-4) = 3 \\
& \qquad \text{thus relax } (3,2) \text{ and put } d(2) = 3, p(2) = 3 \\
& \dots
\end{aligned}
$$

The *first pass* results in distances d and predecessors p:

u	1	2	3	4	5	6
d	0	3	7	12	11	9
p	0	3	1	5	3	3

In the *second pass* we find that out of 10 arcs only 5 are still eligible for relaxation, viz.,

$$(2,4),\ (2,5),\ (2,6),\ (5,4),\ (5,6)$$

The second pass yields distances and predecessors:

u	1	2	3	4	5	6
d	0	3	7	7	6	8
p	0	3	1	5	2	2

(2.2)

Passing through the arc list a third time we find that no more arcs are eligible for relaxation, therefore we stop here. The distances and predecessors (2.2) found during the second pass are optimal.

Now let's see what we have found: column $u = 4$ (2.2) tells us that a shortest path from 1 to 4 has length $d(4) = 7$. Also, a shortest path $1 \to 6$ has length $d(6) = 8$, etc.

But, *what is the shortest path* to, say, $u = 4$? This path can be found by means of the p-row in table (2.2). Let $P_{1,4} = [1, \ldots ? \ldots 4]$ denote the shortest path from $1 \to 4$.

In column $u = 4$ we have $p(4) = 5$. Thus in the shortest path $1 \to 4$ the vertex visited *immediately* before 4 is vertex 5, so

$$P_{1,4} = [1, \ldots ? \ldots 5, 4]$$

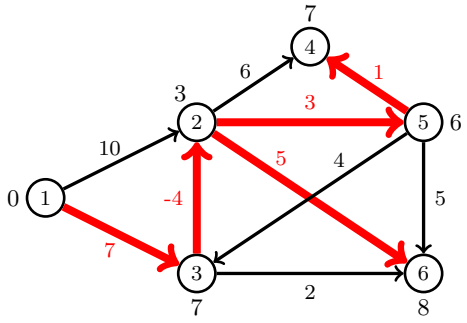
The vertex visited before $u = 5$ is the predecessor $p(5) = 2$. The predecessor of 2 is $p(2) = 3$, the predecessor of 3 is $p(3) = 1$, Here we may stop, because 1 is the starting vertex, thus a shortest path from $1 \to 4$ is

$$P_{1,4} = [1, 3, 2, 5, 4]$$

But we have found even more! Looking closer to the predecessor vector p,

u	1	2	3	4	5	6
$p(u)$	0	3	1	5	2	2

we see that the columns $u = 2, 3, 4, 5, 6$ determine arcs of the form $(p(u), u)$. They form a subset $B \subset A$ of the arcs of our graph $G = [V, A]$, a very special subset. We used *some* of them to determine of shortest path $1 \to 4$. Let us redraw the diagram of our graph and emphasize these arcs, also add the calculated distances $d(u)$ to the vertices of the graph:



The arcs in subset B together with the original vertex set $V = [1, 2, 3, 4, 5, 6]$ form a *subgraph* $T = [V, B]$ of $G = [V, A]$. It has two very special properties:

- T is a *tree*: in T there is exactly one path from vertex 1 to any other vertex in T.
- It is a *spanning tree* because *all vertices* of G graph belong to this tree.

This spanning tree is called the *shortest path tree* of G rooted in vertex 1. From T we can determine immediately *all shortest paths* from 1 to any other vertex in V. For instance, a shortest path from $1 \to 6$ is $Q = [1, 3, 2, 6]$ and it has length $d(6) = 8$.

2.1.7 The complexity of the Bellman-Ford Algorithm

What is the amount of computational work to be done when solving the SSSP for an arbitrary graph G by means of the Bellman-Ford Algorithm? Of course, this depends on both, the *order* $n = |V|$ of G and its *size* $m = |A|$. Let us perform a *worst case analysis*.

At the heart of the Bellman-Ford Algorithm there is the relaxation part:

```
if d(v) > d(u) + w(u,v)
   d(v) = d(u) + w(u,v)
   p(v) = u
```

The CPU-time t required to perform the relaxation of an arc (u, v) certainly depends on computer architecture and processor. This time is bounded by some constant a, i.e., $t \leq a$, where the constant a is machine dependent. However, in any case a is independent of n and m.

The graph has m arcs, so the time required to perform one pass through all arcs cannot be larger than $a \cdot m$. If we cannot exclude *apriori* the existence of a negative cycle, then n passes are necessary. It follows that in total Bellman-Ford requires computing time not more than $a \cdot m \cdot n$. Thus, if we denote by τ the running time of the Bellman-Ford Algorithm, we have in the worst case $\tau \leq a \cdot m \cdot n$, where a is a constant independent of m and n. Mathematicians have a nice formalism to express this bounding, they say that τ is a *big-Oh* of $m \cdot n$, written as

$$\tau = O(m \cdot n), \tag{2.3}$$

This is the *time complexity* of the Bellman-Ford algorithm.

The big O's follow some rather weird arithmetic rules, but as it is a really important concept, so you should learn how to handle it. Chapter 9 of Graham, Knuth, and Patashnik (2003) is a wonderful source and very helpful.

The upper bound (2.3) looks somewhat harmless but it is not. Size m and order n of a graph are not really independent. The number of arcs m may be any number between $m = 0$ (the graph has no arcs at all) to $m = n(n-1)$. In the latter case there is an arc between any pair of vertices, such a graph is called *complete*. These are two extremes, of course. If m is of order $O(n^2)$, i.e. $m \leq Mn^2$, M independent of n, then the graph G is said to be *dense*, it has really a lot of arcs. If m is of order $O(n)$ then we say that G is a *sparse* graph, there are about as many arcs as there are vertices. Practical experience tells us that most graphs in real world applications of the SSSP are sparse.

To summarize: the running time of the Bellman-Ford alogorithm is of order $O(n^3)$ when G is dense, and is runs considerably faster, namely in $O(n^2)$ time when G is sparse.

So, after all, is Bellman-Ford an efficient algorithm ?

Under particular circumstances (negative weights, possibility of negative cycles) it is fairly efficient. But you will find out that for graphs having certain special properties Bellman-Ford is rather slow compared to other algorithms for the SSSP, see Section 2 below.

After having read this *Invitation* so far the question arises:

Are you still interested in this topic?

If so, fine! Welcome on board! Please read on and see what I want from you.

2.2 Where to go from here

There are a few points which I find you should discuss carefully, *issues of general interest*. I have also collected some ideas and suggestions, *optional issues*, which you may find worth to be explored and presented in your thesis. But, of course, you are free (and strongly encouraged) to formulate your own ideas and make them part of your work.

Regarding structure and design: Your thesis should be a *fine mixture of theory and practice*. Develop theoretical concepts and underpin these by appropriate examples. Note that this makes it necessary to implement algorithms in some computing environment. Regarding the latter there are practically no restrictions. The computing environment may be `R`, `matlab/octave`, `java`, `python` or some other programming language.

2.2.1 Issues of general interest

Data structures

A most important point to be discussed quite early in your thesis is how to represent graphs numerically. You will have to learn about *adjacency matrices* and *adjacency lists*. Explain these concepts, discuss their implementation, their advantages and disadvantages, see Cormen et al. (2001, chapter 23) or Gondran and Minoux (1995, chapter 1).

Ralaxation

Provide a more detailed discussion of the relaxation principle. It has some interesting properties which deserve a presentation, see Cormen et al. (2001, chapter 25). Also, give a complete proof that relaxation converges to the optimum and actually finds all shortest paths from a single source vertex. Good references

in this context are Sedgewick and Wayne (2011, chapter 4) and Gondran and Minoux (1995, chapter 2).

The Bellman-Ford Algorithm

Implement the Bellman-Ford Algorithm whose basics we have discussed above. Demonstrate it by running the algorithm on a graph of your choice (may be you find some interesting application). How does your algorithm work in case of a graph having a negative cycle?

Dijkstra's Algorithm

This is one of the most famous and most efficient algorithms for solving the single-source shortest path problem. It is due to Edsger W. Dijkstra (1959)[7]. Recall that in the Bellman-Ford Algorithm we process the list of all arcs *in some order*. However, a clever choice of processing order may improve efficiency a lot. Explain in detail how Dijkstra's algorithm determines the order in which arcs are processed. Show that a naïve implementation has running time $O(n^2)$, where as always $n = |V|$ is the number of vertices. If the graph is *sparse*, then using special data structures this time bound can be improved to $O((m + n) \log n)$, $m = |A|$ the number of arcs. However, Dijkstra's Algorithm can be used only when arc weights are positive. It fails in presence of negative weights, why?

2.2.2 Some more suggestions

Unit weight graphs - Breadth-First Search

There are interesting applications of the shortest path problem where all arcs of the underlying graph have the *same weight*, say $w(u, v) = 1$ for all $u, v \in V$. Find an interesting application of unit weight graphs and give an example. Interestingly, there are some problems in recreational mathematics, which lead to unit weight graphs[8]. The most efficient way to solve the SSSP for these special graphs is a method commonly known as *Breadth-First Search* (BFS). Explain this algorithm, show that its running time is $O(m + n)$. One also says that BFS is *linear in time*. Actually, this is the best we can hope for. The book of Dasgupta, Papadimitriou, and Vazirani (2008, chapter 4) will be very helpful in this context.

[7] (1930–2002), dutch computer scientist

[8] Alcuin of York's (ca. 735-804) medieval puzzle of the goat, the wolfe and the cabbage is a quite famous example, also the Two Jugs or Wine Decanting Problem is a SPP on unit weight graphs.

Longest paths in scheduling

In Example 4 I have presented a classical problem from *scheduling theory*, finding the makespan of a project consisting of several concurrent tasks. The underlying precedence graphs have no cycles and therefore it is always possible to put some order on vertices, a *topological order*. Once such an order is established (you should explain how this can be done), finding the longest path in a precedence graph can be accomplished in *linear time*. Discuss various aspects of the planning problem by means of an interesting example. See Gondran and Minoux (1995, pp. 67). You may also point out the relation to a classical optimization paradigm, *dynamic programming*.

All pairs of shortest paths

There are several interesting problems in combinatorial optimization where it is necessary to determine shortest paths between any pair of vertices in a weighted graph. The most prominent application is the *Traveling Salesman Problem*. A salesman has to visit customers in n different cities which we regard as vertices in a network of roads. How can we design a *shortest route* such that each city is visited *at least once* and the salesman returns to the city where he has started his tour. Note, I have emphasized that each city has to be visited at least once. If it where possible to the design the tour in such a way that each city is visted *exactly once* then the route would by a *Hamiltonian Cycle*. The Hamiltonian Property is very special. Except for special cases of network structure, for instance when the graph is complete, no efficient methods are known to find out whether such a cycle even exists, not to talk about finding it. However, it is possible to *embed* a connected graph into a complete graph which is always hamiltonian. For this to be accomplished, all pairs of shortest paths have to be found.

One reasonable possibility is to apply Dijkstra's Algorithm on each vertex provided there are no negative arc weights. Using a fast implementation of Dijkstra's Algorithm this can be done very efficiently. But there are also other methods of comparable efficiency, the most prominent being the *Algorithm of Floyd-Warshall* which has running time of $O(n^3)$. A nice feature of this algorithm and some of its refinements is that it can handle negative arc weights also. Of course, negative cycles are still not permitted.

Negative arc weights

Negative weights cause problems as we have seen. Not only that they may induce negative cycles which render the SPP unsolvable, they also affect efficiency. Dijkstra runs much faster than Bellman-Ford. So it's quite natural to ask whether there is a way to get rid of negative weights?

The following idea is attractive because of its simplicity and seems so obvious, but it does not work in general. Possible troubles are sketched in Figure 2.7.

In the graph shown on the left side we have a negative weight, $w(3,4) = -9$.

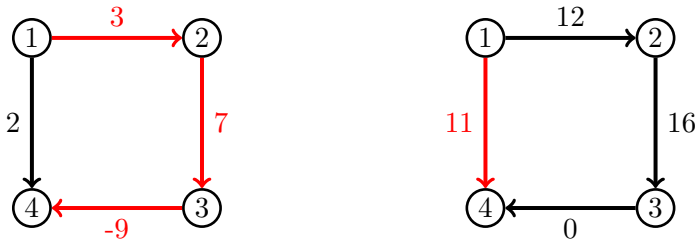


Figure 2.7: Adding constants may cause troubles

A shortest path $1 \to 4$ certainly exists and is easily found by inspection: $P = [1,2,3,4]$. Suppose we add 9 to all weights, then these become nonnegative, but the shortest path changes also, as can be seen from the righthand picture. Now a shortest path $1 \to 4$ is simply $P = [1,4]$.

Although this idea is too simple to work in general, a more sophisticated approach of changing weights does work and give rise to another interesting way of solving the APSP, *Johnson's Algorithm* which on sparse graphs is even more efficient than the algorithm of Floyd-Warshall.

Scaling

Substantial impovements in efficiency are achievable when arc weights are *integers*. This is not an uncommon situation, for instance in scheduling arc weights are processing times which are typically given by positive integers. *Scaling algorithms* take the binary representation of the weights and uncover the bits one at a time from the most significant (leftmost) bit down to the least significant bit. In a first pass shortest paths are determined using the most significant bit of arc weights only, in the second pass we use the two highest order bits, and so on. The point is that these problems are not independent and it is possible to find optimal paths lengths $d_k(u)$ based on the first k bits from the path lengths $d_{k-1}(u)$ very efficiently. This idea has been introducted by Gabow (1985), see also Cormen et al. (2001, pp. 547). Scaling algorithms have a running time of order $O(m \log W)$, where $W = \max w(u,v)$ is the maximum arc weight. Variants of these algorithms can handle also negative arc weights.

2.2.3 To be avoided

Your thesis should not become a new booklet on graph theory. It is not necessary to give an introduction into the basic concepts of graph theory and explain the terms *graph, path, cycle, etc.* You may presuppose that your interested readers either have some knowledge in this field or are willing to aquire it. In the latter case it is sufficient to give some references to books or other ressources of have found to be useful during your studies. See the Annotated Bibliograpy

in Section 3.

Avoid discussion of the SSP on *undirected graphs* unless you know what you are doing. As long as arc weights are positive there is no problem, all standard algorithms will work also for undirected graphs because these can be transformed easily into directed graphs, we have talked about that very briefly on page 30. However, in case of negative weights this simple transformation will inevitably create negative cycles and render the SSP unsolvable. Solving the SSP on undirected graphs with general weights is much more difficult because this requires the concept of *matching*, a very advanced theme in graph theory.

As this thesis topic is one with a strong graph theoretic flavour, you will have to draw diagrams of graphs, there will be tables, outlines and listings of algorithms, etc. *Do not copy such items from anywhere and paste it into your text.* This is very bad style and will not be accepted. As your thesis has to be typeset in LaTeX, watch out for appropriate LaTeX packages. For instance, graph drawing is very easy when you use the `tikz`-package, available from various sites in the web. This packages is also very well documented thus you will have no troubles when drawing fine diagrams. By the way, all graphs in this *Invitation* have been created with `tikz`.

2.3 An Annotated Bibliography

There is a large number of excellent books on graph theory. One of the best introductory texts is Chartrand (1975). It is strongly recommended if you want to get an overview of major concepts of graph theory (which you should). Another (more advanced) book is Berge (1962). Claude Berge (1926–2002) was one of the most influential french mathematicians in the 20th century contributing many deep results in combinatorics and graph theory. Berge was also interested in arts, he was sculptor, painter and novelist[9]. Berge's book has a chapter on shortest paths (chapter 7). In this you can find one of the first formulations of the relaxation pronciple. Interestingly, most books on graph theory do not cover the SSP. An explanation might be that most graph theorists view the SSP not as a problem interesting in graph theory but as one belonging to combinatorial optimization. However, there is a couple of books emphasizing algorithms for graphs and these books usually contain thorough discussions of the SSP. The books I like most are Gondran and Minoux (1995), Christofides (1975) and Gibbons (1991). The text of Michel Gondran and Michel Minoux is strongly influenced by Claude Berge, it has a long chapter on the SSP. There you will find a simple proof that iterated relaxations actually converge to the optimal solution. And you will find there also a detailed acount of the project scheduling problem and longest paths.

In textbooks on combinatorial optimization the SSP always occurs at a promi-

[9] Among other things he wrote the murder mystery *Who killed the Duke of Densmore?* Sherlock Holmes could answer this question by means of a theorem on *perfect graphs*, an important topic in the scientific work of Berge.

nent place as it is *the classical* combinatorial optimization problem. A wonderful book is Lawler (2001). This text is more or less selfcontained, it has a very readable introductory chapter on graph theory (chapter 2) and gives a deep coverage of the SSP in chapter 3. The approach to the SSP is somewhat different from ours, as it is not directly based on the relaxation idea but on a fundamental system of *nonlinear* equations arising from *Bellman's Optimality Principle* which lies at the heart of *dynamic programming*. When talking about combinatorial optimization one must also mention Papadimitriou and Steiglitz (1998) as it is a classical textbook in this field. The SSP occurs in this book at various places (there's no chapter devoted exclusively to the SSP). Really interesting is the formulation of the SSP as a *linear programming problem* with integer-valued decision variables. When you have some background in linear programming, you may find this book a valuable source.

The SSP is also of some significance in computer science, so it does not come as a surprise that textbooks from this field usually contain interesting material. There are three books I strongly recommend. On the first place there is the *Bible Of Computer Programmers*, Cormen et al. (2001). It is a quite voluminous book with an extensive part on algorithms for graphs. There you can find among other things a thorough discussion of the relaxation principle. A fine text is Dasgupta, Papadimitriou, and Vazirani (2008), a really gentle introduction to algorithms with a careful coverage of the SSP. Last, but not least, there is the excellent book of Sedgewick and Wayne (2011). Like Cormen et al. it has a part in algorithms for graphs with a fine chapter on the SSP. This book also discusses in detail the computer implementation of algorithms, the language used throughout the book is `java`.

Finally, regarding the origins of the SSP in the 1950s, you may consult the paper of Schrijver (2012).

2.4 References

[1] C. E. Berge. *The Theory of Graphs and its Applications*. Methuen, London, 1962.

[2] Gary Chartrand. *Introductory Graph Theory*. Dover Publications, 1975.

[3] Nicos Christofides. *Graph Theory - An Algorithmic Approach*. Academic Press, 1975.

[4] Thomas H. Cormen et al. *Introduction to Algorithms*. 2nd. McGraw-Hill Higher Education, 2001.

[5] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh Vazirani. *Algorithms*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2008. URL: http://cseweb.ucsd.edu/~dasgupta/book/toc.pdf.

[6] Edsgar W. Dijkstra. "A note on two problems in connexion with graphs". In: *Mumer. Math.* 1 (1959), pp. 269–271.

[7] L. R. Ford. *Network Flow Theory*. RAND Corporation, 1956.

[8] Harold N. Gabow. "Scaling algorithms for network problems". In: *Journal of Computer and System Sciences* 31.2 (1985), pp. 148–168.

[9] Alan Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1991.

[10] Michel Gondran and Michel Minoux. *Graphs and Algorithms*. John Wiley and Sons, 1995.

[11] L. Graham Ronald, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. 2nd ed. Addison-Wesly, 2003.

[12] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Dover Publications, 2001. URL: www.plouffe.fr/simon/math/CombinatorialOptimization.pdf.

[13] Christos H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Dover Publications, 1998.

[14] Alexander Schrijver. "On the history of the shortest path problem". In: *Documenta Mathematica* (2012), pp. 155–167.

[15] Robert Sedgewick and Kevin Wayne. *Algorithms, 4th Edition*. Addison-Wesley, 2011.

Topic 3

The Seven Bridges of Königsberg

This question is so banal, but seemed to me worthy of attention in that neither geometry, nor algebra, nor even the art of counting was sufficient to solve it.

Leonhard Euler, March 13, 1736 (in a letter to Giovanni Marinoni)

Keywords: *graph theory, Euler paths, Euler cycles, the Chinese Postman Problem and its variants, urban operations research*

3.1 An Invitation

3.1.1 Euler's 1736 paper

This topic deals with a classical problem from graph theory which is very easy to state and also easy to understand. In pictorial language: consider a network of roads and find a tour through the network such that each road is used exactly once. Such a tour, when it exists, will be called an *Euler cycle*, if the tour returns to the location where it started. If initial point and end point of the tour do not coincide but still all raods are traversed exactly once, then we shall call it an *Euler path*.

Leonhard Euler
(1707-1783)

The problem has its origin in a famous puzzle which attracted Euler's interest in 1736. The results of Euler's efforts to solve the puzzle were compiled into one of Euler's most celebrated publications (Euler, 1736). This paper marks the beginning of two new branches of mathematics: *topology* and *graph theory*. Topology, today a basic discipline of modern mathematics, deals with properties of space and bodies which remain invariant with respect to continuous transformations, like stretching, etc. Topology goes back to Gottfried Leibniz (1646–1716) who envisaged a new kind of analysis, a *geometry of position* (*geometria situs*). On the other hand, graph theory deals with *binary*

relations between elements of a given set. As such it is a prominent part of what is known today as *discrete mathematics*.

There is some confusion with respect to the date of publication of Euler's paper. In the records of the St. Petersburg Academy of Science it is noted that Euler presented the Königsberg problem and its solution in a talk given on August 26, 1735. The paper is contained in the *Commentarii* of 1736, but their publication was delayed so that this volume did not appear in print before 1741. Very likely, the presentation date 1735 is a misprint (Grötschel and Yuan, 2012).

3.1.2 Königsberg and a puzzle

In the 18th century Königsberg (Regiomonti), the old capital of East Prussia, was one of the most important cultural and economic centers around the Baltic Sea. It was home of the *Albertina University*, home of the philosopher Immanuel Kant and of several famous mathematicians, just to mention Gustav J. Jacobi, David Hilbert and Hermann Minkowski. Over centuries Königsberg grew into a large and wealthy town with many beautiful churches and a fine cathedral dedicated to Virgin Mary and St. Adalbert. In the midth of the 18th century (the time our story plays) Königsberg had about 50 000 inhabitants. Its situation on the Pregel River made it an ideal trading center for many commodities, such as grain, potash, salt, hemp, and wood (R. J. Wilson, 1986). The city was set on both sides of the Pregel, and included two large islands which were connected to each other and the mainland by seven bridges: Grüne Brücke, Krämer Brücke, Schmiedebrücke, Hohe Brücke, Holzbrücke, Köttelbrücke, and Honigbrücke.

Königsberg arround 1930
In the foreground the open Schmiedebrücke

In Königsberg there was the nice tradtion of *Corso*. On sunday afternoon families used to promenade through the center of the city, having some rest in one of the cafés, meeting friends, having conversation, and, yes, reasoned about a remarkable puzzle which must have originated at that time:

Is it possible to walk through the city, crossing every of the seven bridges once and only once and to return to the point where the promenade has started?

3.1.3 Euler takes notice of the puzzle

Nobody could solve this puzzle and since all attempts to solve it had always failed it was commonly believed that this task is impossible. The question re-

Figure 3.1: Map of Königsberg and its seven bridges, Source: Heritage Schoolhouse

mained unsettled until the famous Swiss mathematician Leonhard Euler (1707-1783) took notice of it. It is not know when and from whom Euler learned about this problem for the first time. Actually, Euler never visited Königsberg, as far as we know. But very likely the historical course was this (Sachs, Stiebitz, and J. R. Wilson, 1988): Carl Leonhard Gottlieb Ehler, at that time mayor of the city of Dantzig was a friend of Euler and a mathematical enthusiast. During the years 1735-1742 he corresponded with Euler in St. Petersburg, mainly acting as an intermediary between Euler and Heinrich Kühn, professor of mathematics at the Academic Gymnasium in Dantzig. Via Ehler Kühn communicated the Königsberg Problem to Euler. In a letter dated from March 9, 1736, Ehler wrote to Euler[1]:

> *You would render to me and our friend Kühn a most valuable service, putting us greatly in your debt, most learned Sir, if you would send us the solution. which you know well, to the problem of the seven Königsberg bridges, together with a proof. It would prove to be an outstanding example of the calculus of position [Calculi Situs], worthy of your great genius. I have added a sketch of the said bridges ...*

It took Euler only a few days to find a solution. He immediately reported it to Giovanni Jacobo Marinoni (1670–1755), astronomer at the court of Emperor Leopold I. in Vienna. In a letter dated from March 13, 1736, Euler wrote:

> *A problem was proposed to me about an island in the city of Königsberg, surrounded by a river spanned by seven bridges, and I was asked whether someone could traverse the separate bridges in a connected walk in such a way that each bridge is crossed only once. I was informed that hitherto no-one had demonstrated the possibility of doing this, or shown that it is*

[1]This and the following excerpts of letters are translations taken from Sachs, Stiebitz, and J. R. Wilson (1988).

> *impossible. This question is so banal, but seemed to me worthy of attention in that neither geometry, nor algebra, nor even the art of counting [ars combinatoria] was sufficient to solve it. In view of this, it occurred to me to wonder whether it belonged to the geometry of position [geometria situs], which Leibniz had once so much longed for. And so, after some deliberation, I obtained a simple, yet completely established, rule with whose help one can immediately decide for all examples of this kind, with any number of bridges in any arrangement, whether such a round trip is possible, or not...*

And on April 13, 1736 Euler replied to Ehler's letter of March 9. The following citation reveals clearly that Euler didn't consider the Königsberg problem an interesting one from the standpoint of a mathematician. Indeed, he considered it merely a *puzzle*:

> *Thus you see, most noble Sir, how this type of solution bears little relationship to mathematics, and I do not understand why you expect a mathematician to produce it, rather than anyone else, for the solution is based on reason alone, and its discovery does not depend on any mathematical principle. Because of this, I do not know why even questions which bear so little relationship to mathematics are solved more quickly by mathematicians than by others. In the meantime. most noble Sir, you have assigned this question to the geometry of position, but I am ignorant as to what this new discipline involves, and as to which types of problem Leibniz and Wolff*[2] *expected to see expressed in this way.*

Not much later Euler must have changed his mind, now considering this puzzle important enough to write a paper about it.

3.1.4 Euler's solution

In the sequel I will cite freely from Euler's paper and the translation due to Michael Behrend (2012). Euler starts his paper by relating the Königsberg Problem to the geometry of position (*geometria situs*) initiated by Leibniz.

And now let us listen to Euler's own words. In paragraph 2 he writes:

§ 2. This problem, then, which was described to me as quite well known, was as follows: At Königsberg in Prussia there is an island A called der Kneiphof, and the river around it is divided into two branches, as shown in the figure; and the branches of this river are crossed by seven bridges a, b, c, d, e, f, and g. The following question was now raised concerning these bridges: whether someone could arrange a walk in such a way as to travel over every bridge once and not more than once. Some people (I was told) deny that this is possible, and others doubt it; but nobody asserts it. From this I set myself the following quite general problem: whatever the form of the river and its

[2] Abraham Wolff (1710–1795), jewish mathematician in Berlin, friend of Euler. Gotthold Ephraim Lessing memorialized Wolff by the figure of the dervish Al Hafi in his drama *Nathan der Weise*.

distribution into branches, and whatever the number of bridges, to find whether it is possible for each bridge to be crossed only once, or not.

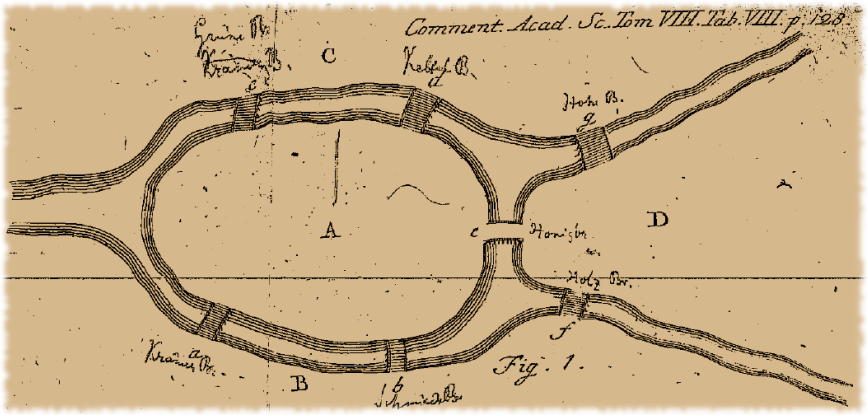


Figure 3.2: *A simplified map of Königsberg*

Euler begins his analysis by replacing the map of the city by a simpler diagram, as it is shown in Figure 3.2.

It is likely that for his study Euler used an even more simplified diagram which represents what is called a *graph*.

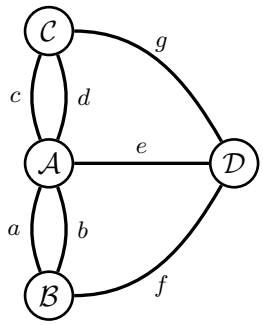


Figure 3.3: *The translation of Figure 3.2 into a graph*

You can find a more formal definition of graphs in Topic 2: *Shortest Paths in Networks*. However, for our purpose it is sufficient to pursue a somewhat pedestrian-like approach: A graph is a pair $G = (V, E)$ where V is a set of points (in our case $V = \{A, B, C, D\}$, the islands and riverbanks) and E a set of lines connecting these points (the bridges $E = \{a, b, c, d, e, f, g\}$). The points are called *vertices*, the line segments *edges*.

Euler then remarks that in principle it is possible to solve the Königsberg Problem by making an exhaustive list of all possible routes and finding by inspection whether a particular route satisfies the conditions of the problem. But he immediately rejects this approach as impractical because the number of different routes will be too large, in general. Indeed, Euler doesn't want to solve only the

Köngsberg Problem, he is seeking a general method suitable for much bigger networks.

Next Euler represents routes as sequences of capital letters. For instance $ABDC$ represents the route starting on the island A, then passing to the riverbank B by using bridge a or b, continuing to island D crossing the bridge f and going to C by passing bridge g.

In modern graph-theoretic terms such a sequence is called a *walk*. Observe that Euler when writing $ABDC$ for a walk allows for some ambiguity in that the used bridges are not specified. Later in his paper he resolves this ambiguity by writing more explicitely for a possible walk from A to C:

$$AaBfDgC$$

More generally: If a graph has vertex set V and edge set E, then a walk is a sequence of alternating vertices and edges $v_0, e_1, v_1., \ldots, e_n, v_n$. If all edges are different (which is certainly required by the Königsberg Problem) the walk is called a *path*. If initial and terminal vertex of a path coincide, the path becomes a *cycle*. If this cycle contains all edges of the graph the path (cycle) is called *Euler path (cycle)*.

By careful analysis Euler identifies the crucial condition for the existence of a solution of the Königsberg Problem and its generalizations to more complex networks. It is the *parity (odd or even) of the number of bridges* which lead to an area. He writes in paragraph 20:

§ 20. Therefore in any given case it will be very easy to decide straightaway whether a crossing by each bridge once only can be planned or not, with the help of the following rule. If there are more than two regions with an odd number of bridges leading into them, then it can safely be stated that there is no such crossing. And if there are exactly two regions with an odd number of bridges leading into them, then the crossing can be done, provided the walk is started in one of these two regions. Finally, if there is no region at all with an odd number of bridges leading to into it, then the crossing can be planned in the desired way, and the start of the walking can be placed in any region. Therefore the rule just given fully satisfies the statement of the problem.

In modern graph theory, the number of edges leading to a vertex v is called the *degree* of that vertex. It is usually denoted by $d(v)$. Thus **§ 20** may be summarized by the statement: let $G = (V, E)$ be a graph which is *connected*, which means, that it is possible to find a path between any pair of vertices, so there are no isolated areas. Then:

1. G contains an Euler cycle, if there are no vertices of odd vertex degree.
2. It contains an Euler path (initial and terminal vertex are different), if G has at most two vertices of odd degree.

These conditions are really easy to check. For the Königsberg Problem we read off from Figure 3.3:

$$d(A) = 5, \quad d(B) = 3, \quad d(C) = 3, \quad d(D) = 3,$$

so, all vertex degrees are odd numbers, therefore the Königsberg Problem has no solution. There is no Euler cycle and also no Euler path.

Euler discusses also another example, where an Eulerian path exists but no Euler cycle. See Figure 3.4.

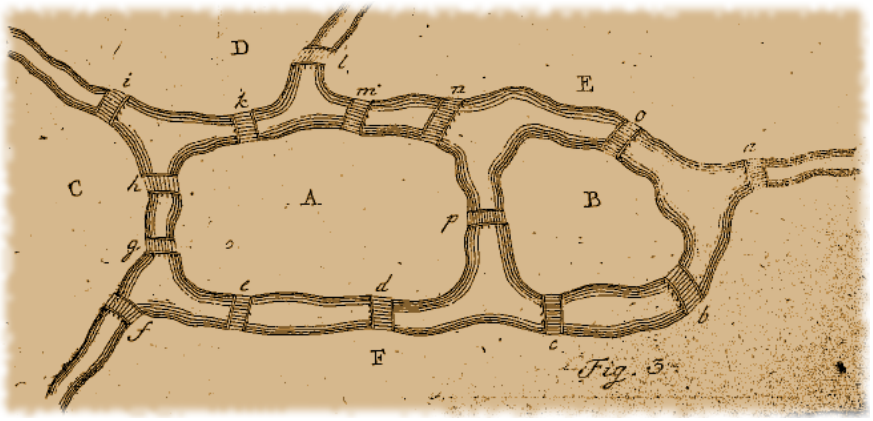


Figure 3.4: *Another example in Euler's paper*

Representing areas again as vertices, it is an easy task to draw the graph representing this map. Counting the vertex degrees yields:

$$d(A) = 8, \quad d(B) = 4, \quad d(C) = 4, \quad d(D) = 3, \quad d(E) = 5, \quad d(F) = 6$$

There are exactly two vertices of odd degree, D and E, thus no Euler cycle exists, but an Euler path can be found. It must lead from D to E (or *vice versa*) such that it traverses each bridge exactly once.

Can you find this path?

Of course, you can find *one* such path (may be more?) just by patiently searching in the graph for an Eulerian path. But, certainly, you will come to the conclusion, that a naïve search may be too cumbersome in bigger graphs. What is needed, is an *algorithm*! Let us read what to Euler says about this problem. In paragraph 21 (the last in his paper) he writes:

§ 21. But when it has been found that such a crossing can be arranged, the question remains how the walk is to be carried out. For this I use the following rule: in imagination, let the bridges that lead from one region to another be removed in pairs as many times as possible, by which the number of bridges will in most cases be greatly reduced; then let a walk be planned across the remaining bridges, which is easily done; then the bridges that were imagined as removed will not much disturb the walk so found, as will at once be seen on very little consideration; nor do I think it necessary to give more rules for arranging the walks.

Frankly speaking, the rule Euler formulates can hardly be considered an algorithm. Still, there is no doubt that Euler knew how to construct an Eulerian path. But we should bear in mind that the theory of algorithms did not exist

in Euler's time nor did Euler have the concept of *recursion* which proves to be very useful (Grötschel and Yuan, 2012).

3.1.5 What happened to the problem later?

About 100 years after their publication in the *St. Petersburg Commentarii* Euler's discoveries about the Königsberg Problem were almost forgotten. In 1851 E. Coupy published a french translation of Euler's paper, mainly intended for his students at the *École Polytechnique* in Paris, and in 1875 Louis Saalschütz, professor at the Königsberg University pointed at, that after a new bridge (*Kaiserbrücke*) has been built to connect riverbanks $\mathcal{B}$ and $\mathcal{D}$ an Euler Path from $\mathcal{A}$ to $\mathcal{D}$ was now possible: Hence, there are only two odd vertices, $\mathcal{A}$ and

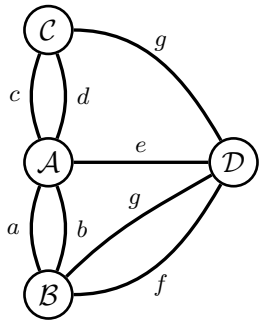


Figure 3.5: *In 1875 a new bridge has been built connecting $\mathcal{B}$ and $\mathcal{D}$*

$\mathcal{C}$, so at least one Euler path from $\mathcal{A} \to \mathcal{C}$ is possible. E.g.,

$$[\mathcal{A}, \mathcal{D}, \mathcal{B}, \mathcal{A}, \mathcal{B}, \mathcal{D}, \mathcal{C}, \mathcal{A}, \mathcal{C}]$$

However, an Euler cycle is still impossible.

Euler's findings have been rediscovered several times notably in *Recreational Mathematics*[3], in particular in the context of *unicursal problems*, *mazes* and *labyrinths*.

An *unicursal problem* is a diagram tracining puzzle: we are given a diagram like that the famous Lantern of Santa Claus in Figure 3.6. It is required to draw the diagram with as few as possible pen strokes without drawing a line more than once. The lantern is certainly unicursal, as it has an Euler path from $A \to B$. Indeed, there are 44 such paths!

3.1.6 An epilog: Königsberg and its bridges today

The beautiful medieval city of Königsberg no longer exists. It has been destroyed by bombing raids in late summer 1944 and military actions during the battle of Königsberg in winter 1944/45. The bridges, the principal actors in

[3] You may have seen that Chapter 1 in our booklet is devoted to this subject.

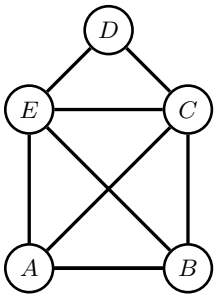


Figure 3.6: *The Santa Claus Lantern is unicursal*

our story, were either destroyed or severly damaged. Today Königsberg is a Russian naval base named Kaliningrad, the river Pregel has been renamed to Pregolya. The seven bridges no longer exist. Today there are eight bridges crossing the Pregolya river, partly reconstructions of the old bridges. From the Schmiedebrücke only its pillars are still existing.

3.1.7 The Chinese Postman Problem

Are there any *useful applications* of Euler's findings? So far, you might have got the impression that Euler has merely solved a *puzzle*, not more. Yes, there are! Let me introduce you to a most famous optimization problem: the Chinese Postman Problem.

In the city of Qufu[4] (Shandong Province, China) there are many post offices. Each post office serves a certain district of the city, and these districts are divided into subnetworks of roads which are assigned to postmen for mail delivery. Figure 3.7 displays such a subnetwork of roads to be served by a single post-

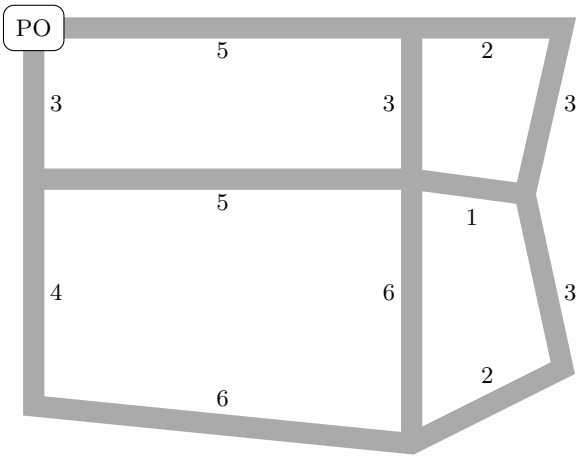


Figure 3.7: A road network for a Chinese postman

[4]Place of birth of the great Chinese philosopher Confucius (551 BC - 479 BC).

man. The numbers attached to the roads are walking distances in kilometers, say. On each of his daily tours the postman starts at the post office (PO), then has to pass through each of the roads *at least once* and finally returns to the office. Now, a natural question is: *How can we arrange a tour of minimum length?* This question has become known as the famous *Chinese Postman Problem* (CPP), stated and solved first by the Chinese mathematician Mei-Ko Kwan in 1960. The name of this problem is very likely due to Jack Edmonds (1965) who coined the term in honor of Mei-Ko Kwan.

The CPP is a classical *combinatorial optimization problem* like the *Traveling Salesman Problem* (TSP), but unlike the latter, it is well-behaved in the sense that even large instances of this problem can be solved in reasonable time.

But, *how is the CPP related to the Königsberg Puzzle discussed by Euler?*

To see this relation, let us first represent the road network of Figure 3.7 schematically by a graph. The post office, junctions and turns are the vertices of the graph, the roads are repesented by edges which are assigned as weights the distances. The resulting graph is shown in Figure 3.8 below, where the post office is located in vertex 1.

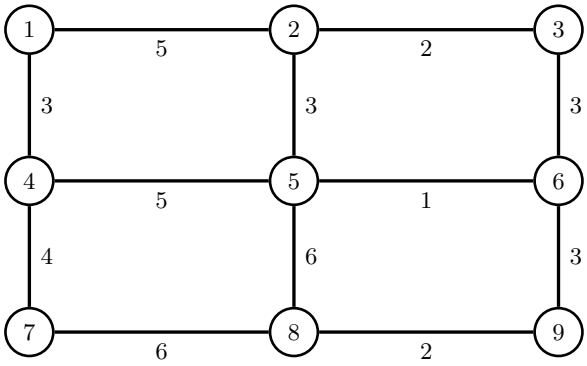


Figure 3.8: The graph corresponding to the network of Figure 3.7

If this graph had an Euler cycle this would immediately yield a solution of the CPP: the postman simply had to follow the cycle, and as the cycle passes through each edge it is certanly of minimum length because any Euler cycle has the same length, just the sum of all edge lengths of the graph. In this case, there is nothing to optimize.

But looking at Figure 3.8 you realize that this graph is not Eulerian, because the vertices 2, 4, 6 and 8 have odd degree. Therefore no Euler cycle can exist.

So, what should the postman do? Easy, think in practical terms: *he will have to traverse some of the streets more than once.* And now, we do indeed have an optimization problem: determine those streets which have to be traversed twice in such a way that the resulting round trip has minimum length.

Mei-Ko Kwan's solution approach is *in principle* a very easy one:

- Determine all vertices of odd degree in the graph. Let this set be denoted by M. The number $m = |M|$ of elements in M will always be an even number, as Euler (1736) shows in § 16 of his paper. In our example $M = \{2, 4, 6, 8\}$.
- Find all *pairwise matchings* of the vertices in M. In other words, find the set of all pairings of elements in M. If $|M| = m$, then it is easily seen that the number of pairwise matchings equals $(n-1)(n-3)\cdots 3 \cdot 1$. In our case we have $m = 4$ and therefore we have $3 \cdot 1 = 3$ matchings:

$$M_1 = \{2 \leftrightarrow 4, 6 \leftrightarrow 8\}, \quad M_2 = \{2 \leftrightarrow 6, 4 \leftrightarrow 8\}, \quad M_3 = \{2 \leftrightarrow 8, 4 \leftrightarrow 6\}$$

- For each matching determine the *shortest paths* connecting the vertices of each pair:

Matching	vertices	shortest path	length
M_1	$2 \leftrightarrow 4$	$[2, 1, 4]$	8
	$6 \leftrightarrow 8$	$[6, 9, 8]$	5
Total length of M_1			13

Note that the shortest path $2 \leftrightarrow 4$ is not unique. We could have taken as well the path $[2, 5, 4]$ also having length 8.

Similarly,

Matching	vertices	shortest path	length
M_2	$2 \leftrightarrow 6$	$[2, 5, 6]$	4
	$4 \leftrightarrow 8$	$[4, 7, 8]$	10
Total length of M_2			14

And finally,

Matching	vertices	shortest path	length
M_3	$2 \leftrightarrow 8$	$[2, 5, 8]$	9
	$4 \leftrightarrow 6$	$[4, 5, 6]$	6
Total length of M_3			15

Matching M_1 has the shortest total length, so duplicate the edges of the two shortest paths comprising this matching. In other words, for the shortest path $[2, 1, 4]$ duplicate edges $(2, 1)$ and $(1, 4)$, for the path $[6, 9, 8]$ add edges $(6, 9)$ and $(9, 8)$. These are the roads the postman has to walk twice.

The last step is to figure out an Euler cycle in the augmented graph. As this network is a really small one, we may do this by simple inspection. Just take a pencil and pass along the edges of the graph depicted in Figure 3.9 recording the vertices visited to obtain:

$$C = [1, 2, 3, 6, 9, 6, 5, 2, 1, 4, 5, 8, 9, 8, 7, 4, 1]$$

We note in passing that this solution is *not unique*.

This was a very simple example, a toy problem, as I have to concede. But the CPP isn't merely a toy problem, it has many serious and important applications. And in these routing has to be performed in networks of really large size. For this task we need an algorithm, of course. Actually, we need *three* algorithms:

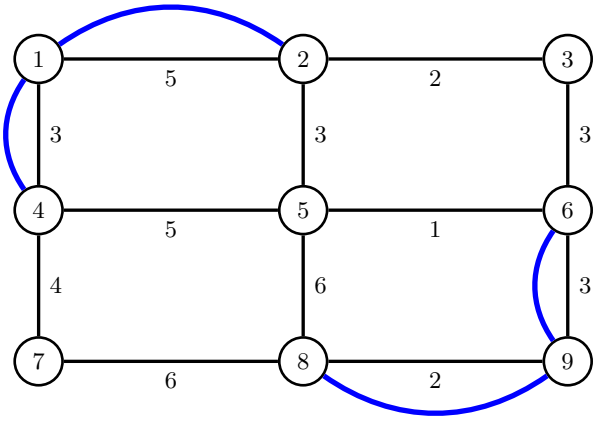


Figure 3.9: The augmented graph

1.) An algorithm for determining *shortest paths* in a weighted graph. *Floyd's Algorithm* mentioned in Topic 2 (*Shortest Paths in Networks*) is a good choice.

2.) We need an algorithm to solve the *minimum cost matching* of odd-degree vertices.

3.) Finally, we need one more algorithm to determine an Euler cycle in the augmented graph.

In real world applications of the CPP networks sometimes have hundreds or even thousands of vertices and edges. Clearly, what is needed are *efficient* algorithms to perform the steps outlined above. More about that in Section 2.

Typical examples of large scale CPPs come from *urban operations research*:

- Routing of trucks for waste collection and street cleaning.
- Optimal organization of snow and ice control on roads during winter time.
- Routing of schoolbuses, etc.

Here are some examples from industrial production where the CPP also proves to be very useful:

- In shipbuilding industry huge plates of steel have to be cut, a process usually performed by plasma cutting devices. Here it is necessary to minimize the number of piercing points and waste of raw material.
- In large storage depots and container terminals stacker cranes are used to move goods and containers around. A typical stacker crane must start from an initial position, perform a set of movements, and return to the initial position. The objective is to schedule the movements of the crane so as to minimize total cost.
- Multifunctional robots are key elements of modern industrial production. They must carry out complicated movements depending on the task to be performed. These movements can be optimized in such a way that

processing times and consumption of energy are minimized.

In all these examples (and there are many more!) it has been reported that formulation and solution of the underlying CPPs resulted in impressive gains of efficiency and savings of cost, see Section 3, *An Annotated Bibliography* below for some references.

3.2 Where to go from here

Writing a thesis about Euler cycles and Chinese postmen is a nice challenge as you will certainly realize when you get more and more involved into the subject. On one side there is beautiful mathematics, on the other side there are very interesting applications.

In this section I present you some ideas you make take care of.

3.2.1 Issues of general interest

At the outset, however, an important point: do assume that the graphs you are dealing with are *connected*. Roughly, this means that there exist paths between any pair of vertices in a graph.

You will have to state this assumption, but I recommend not to put too much emphasis on this issue. This would lead you too far afield as connectivity is a nontrivial graph property, in particular in case of directed graphs, a concept to be introduced shortly.

Algorithms for finding Euler cycles

As we have remarked above, Euler did no outline a general procedure to determine Euler cycles. This gap was closed more than 100 years after Euler's 1736 paper. Two classical algorithms have been invented during the 19th century, interestingly without reference to Euler's original work on the subject which at that time has been more or less forgotten.

The paper of Carl Hierholzer (1873) presents the first algorithm together with a complete proof of the statement that a connected graph has an Euler cycle *if and only if* all its vertices have even degree. Actually, Euler only proved the *if*-part, i.e., a necessary condition for this property. Hierholzer's approach is simple and elegant. It successively finds cycles in the graph $G = (V, E)$ and glues them together to an Euler cycle. When properly implemented (using a *stack* data structure) this algorithm is very efficient, the amount of computational work to find an Euler cycle is proportional to the number of edges $|E|$.

The other algorithm is due to Fleury (1883). Actually, it is the most often cited algorithm as you can check by a quick web search. Fleury's algorithm is so popular because it is very intuitive: just walk through the graph deleting each edge after it has been traversed, *unless it is a bridge* and you are forced to

walk over it. A bridge is an edge which when removed from the graph results in disconnected components. Whereas it is easy to figure out whether an edge is a bridge in small graphs, this tasks becomes quite formidable in bigger graphs and requires an algorithm, the paper of Jens Schmidt (2012) presents one. As a result the computational complextity of Fleury's algorithm is considerably higher than that of Hierholzer's algorithm.

There exist alternatives to Hierholzer and Fleury, of course. One such alternative is an algorithm due to Tucker (1976). You can find a detailed presentation and analysis of algorithms to find Euler paths and Euler cycles in the profound book of Fleischner (1991, Chapter X.).

Your thesis should discuss these algorithms (at least those of Hierholzer and Fleury) carefully, their pros and cons, their time complexity. Maybe (yet another challenge) you write small computer programs to implement them. You may do this in `R`, `matlab`, its clone `octave` or any other environment you like. Also, you will have to think about practical and efficient ways to represent graphs by appropriate data structures. And, of course, you should illustrate the capabilities of your programs by some nice examples.

The Chinese Postman Problem

The CPP is certainly the most obvious application of Euler cycles. Recall from the *Invitation* that the CPP is a combinatorial optimization problem that consists of three layers: at the top level an Euler cycle is sought in a graph which has been suitably augmented. This augmentation, the addition of edges representing those which have to be traversed more than once, requires a shortest path problem and a minimum cost matching problem to be solved.

For the shortest path problem you may have a look into Topic 2 for a first orientation. The book of Christofides (1975) provides more technical information about shortest paths and about the matching problem. Regarding the matching problem: this is really difficult. Therefore I recommend to explain briefly the basics and avoid getting too much involved into the general matching problem. Just take it as some kind of a blackbox. When you implement the CPP then use any of the available software libraries to handle the matching part of the CPP.

It is a very good idea to illustrate your exposition of the CPP by one or more well-chosen applications. An easy-to-read and very informative first introduction can be found in chapter 6 of Larson and Odoni (1981). Material useful in this context is contained in chapters 7 and 16 of Farahani and Miandoabchi (2013).

A class of important applications is the provision of public services like municipal waste collection. The paper of Belien, De Boeck, and Van Ackere (2014) is an up to date exposition with many references which you may find very helpful.

3.2.2 Some more suggestions

Directed graphs

In a directed graph all edges have an *orientation*, i.e., using once again the analogy of a network of roads, all roads are one-way. Such directed edges are usually called *arcs*. Also directed graphs can have Euler cycles. Figure 3.10 shows you an example of a directed graph which has an Euler-cycle. You

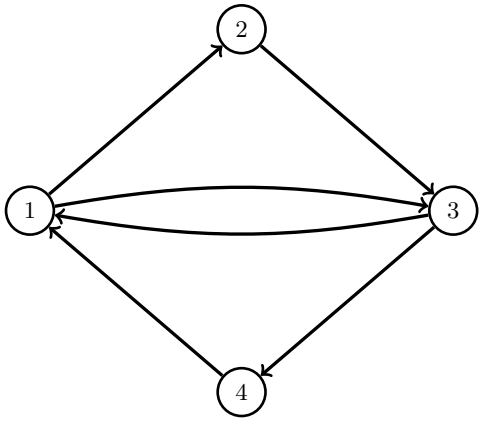


Figure 3.10: An Eulerian digraph

should formulate conditions for a directed graph to have such cycles (or paths) and devise algorithms for finding them. Do the algorithms of Hierholzer and Fleury still work in case of directed graphs?

Once again: the Chinese postman

The CPP can also be considered in directed graphs. Now the situation is somewhat different compared to the CPP in undirected graphs. In the latter the postman will never have to walk along a road more than twice (can you explain, why?). In a directed graph, however, it may be the case that the postman has to traverse arcs (one-way roads) several times. On the other hand, the matching problem is easier to solve. It boils down essentially to a classical *transportation* or *minimum cost flow problem*.

Just to get an impression of the situation have a look at Figure 3.11. It displays a small network with 13 vertices consisting entirely of one-way streets, distances are given in units of 100 m. You can easily check that this directed graph is not Eulerian, but again an augmentation can establish this property. The augmentation process results in the Eulerian network shown in Figure 3.12. The street connecting vertices 6 and 10 has to be traversed five times! The optimal postman tour has total length 10 160 m and is found to be

$$C = [1, 2, 3, 6, 10, 11, 12, 13, 6, 10, 7, 9, 11, 12, 13, 6, 10, 7, 5,$$
$$4, 8, 9, 11, 12, 13, 6, 10, 7, 5, 2, 3, 6, 10, 7, 5, 4, 1],$$

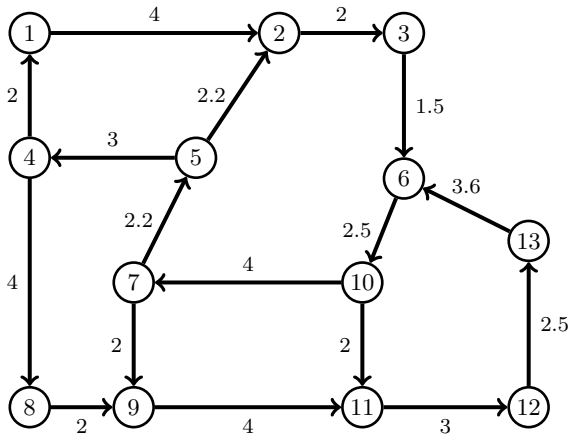


Figure 3.11: A network with one-way streets

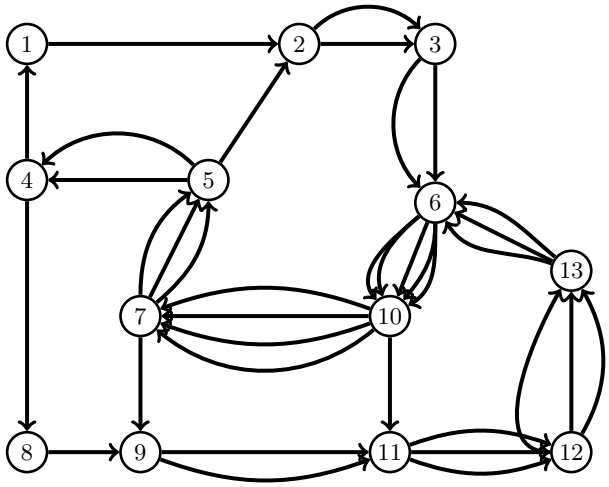


Figure 3.12: The Eulerian network

Can you verify my calculations?

Really poor guys: rural and the windy postmen

The classical CPP is a *tractable problem*, i.e., it can be solved in polynomial time. But there are variants of considerable importance in practical applications that are really hard. One such variant is the *Rural Postman Problem* (RPP). Here the network consists of two types of roads: there are roads whose traveral is mandatory (deliver mail) and other roads which *may* be traversed but need not. A typical example is displayed in Figure 3.13 where the mandatory roads are drawn in heavy lines. We may think of a postman who has to serve two districts which are connected by roads belonging to a district served by some other postman. Again an augmentation process is necessary, but the difficulty is

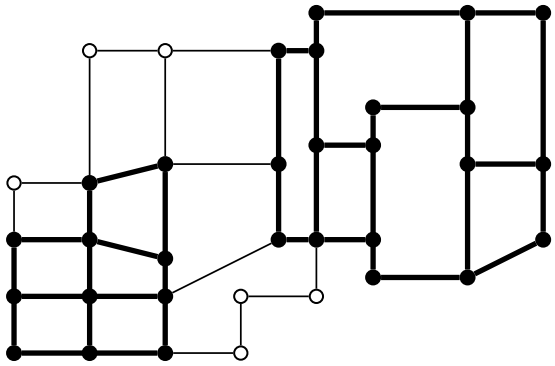


Figure 3.13: A network for a rural postman

that for this process roads of both types (mandatory and non-mandatory) may be used. The RPP is indeed an extremely difficult combinatorial optimization problem.

Finally, there is the *Windy Postman Problem*. This poor guy has to serve a network of roads which may be traversed in any direction (no orientation), but the cost of traversal does depend on the direction, see Figure 3.14 Think of a

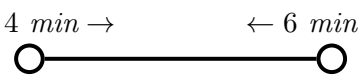


Figure 3.14: Direction dependent costs for the windy postman

postman delivering his mail by using a bicyle and edge weights are travel times. If there is some headwind then these travel times certainly depend on direction. Again, no efficient solution procedure is known. A thorough discussion of these problems is found in the papers of Eiselt, Gendreau, and Laporte (1995a) and (1995b).

3.3 An Annotated Bibliography

To prepare your thesis you will need certainly some acquaintance with basic terminology and concepts from graph theory. Unfortunately, terminology in graph theory is far from being standardized, see footnote 3 in Topic 2. Anyway, gentle introductions to graph theory are the books by Chartrand (1975), Hartsfield and Ringel (2003) and Gibbons (1991). Gibbons's book is more technical in that it puts strong emphasis on algorithms, still it is very readable. In chapter 6 you find a thorough discussion of Eulerian graphs (directed and undirected) as well as postmen problems. Christofides (1975) is one of my favorite books. It combines both, a clear exposition of concepts and a detailed discussion of algorithms. Chapter 9 of this book is devoted to the Euler problem, chapter 12 to the matching problem. Another great book is Gondran and Minoux (1995),

in particular chapter 8. This chapter has also some exercises presenting various applications.

Regarding historical background, I recommend the fine textbook Biggs, Lloyd, and R. J. Wilson (2006), which is a commented collection of milestone publication in graph theory over the period 1736–1936. It begins with an English translation of Euler's 1736 paper and translations of several other papers related to the Euler problem. Here you find also interesting facts about Euler cycles and labyrinths! Grötschel and Yuan (2012) gives an account of the history of the Königsberg bridges puzzle, Euler's work on it, Mei-Ko Kwan and the Chinese Postman Problem.

The classical paper on the CPP is certainly Edmonds and Johnson (1973). At the heart of this publications there is a thorough discussion of the matching problem which has to be solved for the CPP.

If you want to read more from and about Leonhard Euler, consult The Euler Archive (2011) hosted by the Mathematical Association of America. It contains a small part of Euler's enormous *Opera Omnia*, some of his papers come with English translations.

The paper Hierholzer (1873) has a tragic history. He died in 1871 at an age of only 31 years. The cited paper was published posthumously after it has been prepared without written records by Hierholzer's friends Christian Wiener and Jacob Lüroth. An English translation is given in Biggs, Lloyd, and R. J. Wilson (2006), a text already mentioned.

Janet Heine Barnett (2005) and R. J. Wilson (1986) are especially worth to be studied because both of them contains a thoroughly commented translation of Euler's 1736 paper stating its findings in modern graph theoretic terms.

A detailed presentation of Eulerian graphs is the 2-volumne series Fleischner (1990) and Fleischner (1991). In Chapter X. you will find various algorithms for finding Euler cycles together with a careful discussion of their efficiency.

Finally, regarding the CPP: have a look at Eiselt, Gendreau, and Laporte (1995a) and (1995b). Although these papers are rather technical, take your time and read them, as these papers are consider standard reference texts to the CPP and its hard variants, e.g., the rural postman.

3.4 References

[1] The Euler Archive. *A digital library dedicated to the life and work of Leonhard Euler*. 2011. URL: http://eulerarchive.maa.org/.

[2] Michael Behrend. *Republications in maze mathematics*. 2012. URL: www.cantab.net/users/michael.behrend/repubs/maze_maths/pages/euler.html.

[3] J. Belien, L. De Boeck, and J. Van Ackere. *Municipal Solid Waste Collection and Management Problems: A Literature Review*. 2014. URL: https:

//lirias.kuleuven.be/bitstream/123456789/407421/1/Municipal+Solid+WasteCollectionProblems+-+final+paper_revised_3.pdf.

[4] N. L. Biggs, E. K. Lloyd, and R. J. Wilson. *Graph Theory 1736-1936*. Oxford University Press, 2006.

[5] Gary Chartrand. *Introductory Graph Theory*. Dover Publications, 1975.

[6] Nicos Christofides. *Graph Theory - An Algorithmic Approach*. Academic Press, 1975.

[7] Jack Edmonds. "The Chinese postman problem". In: *Operations Research* 13. Suppl. I (1965), p. 373.

[8] Jack Edmonds and Ellis L. Johnson. "Matching, Euler tours and the Chinese postman". In: *Mathematical Programming* 5.1 (1973), pp. 88–124.

[9] H. A. Eiselt, Michel Gendreau, and Gilbert Laporte. "Arc Routing Problems, Part I: The Chinese Postman Problem". In: *Operations Research* 43.2 (1995), pp. 231–242.

[10] H. A. Eiselt, Michel Gendreau, and Gilbert Laporte. "Arc Routing Problems, Part II: The Rural Postman Problem". In: *Operations Research* 43.3 (1995), pp. 399–414.

[11] Leonhard Euler. "Solutio problematis ad geometriam situs pertinensis". In: *Commentarii Academiae Scientarum Imperialis Petropolitanae* 8 (1736), pp. 128–140. English translation: Michael Behrend. *Republications in maze mathematics*. 2012. URL: www.cantab.net/users/michael.behrend/repubs/maze_maths/pages/euler.html.

[12] R.Z. Farahani and E. Miandoabchi. *Graph Theory for Operations Research and Management: Applications in Industrial Engineering*. Business Science Reference, 2013.

[13] H. Fleischner. *Eulerian Graphs and Related Topics I*. Annals of Discrete Mathematics. North Holland Pub. Col, 1990.

[14] H. Fleischner. *Eulerian Graphs and Related Topics II*. Annals of Discrete Mathematics. North Holland Pub. Col, 1991.

[15] Alan Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1991.

[16] Michel Gondran and Michel Minoux. *Graphs and Algorithms*. John Wiley and Sons, 1995.

[17] M. Grötschel and Y. Yuan. "Euler, Mei-Ko Kwan, Kïigsberg and a Chinese Postman". In: *Documenta Mathematica* Extra Volume ISMP (2012), pp. 43–50.

[18] Nora Hartsfield and Gerhard Ringel. *Pearls in Graph Theory, A Comprehensive Introduction*. Dover Publications, 2003.

[19] Janet Heine Barnett. *Early Writings on Graph Theory*. 2005. URL: www-users.math.umn.edu/~reiner/Classes/Konigsberg.pdf.

[20] Carl Hierholzer. "Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechungen zu umfahren". In: *Mathematische Annalen* (1873), pp. 30–32.

[21] R.C. Larson and A.R. Odoni. *Urban operations research*. Prentice Hall PTR, 1981. URL: http://web.mit.edu/urban_or_book/www/book/.

[22] H. Sachs, M. Stiebitz, and J. R. Wilson. "An Historical Note: Euler's Königsberg Letters". In: *Journal of Graph Theory* 12 (1988), pp. 133–139.

[23] Jens M. Schmidt. *A Simple Test on 2-Vertex- and 2-Edge-Connectivity*. 2012. URL: http://arxiv.org/abs/1209.0700.

[24] Alan Tucker. "A New Applicable Proof of the Euler Circuit Theorem". In: *The American Mathematical Monthly* 83.8 (1976), pp. 638–640.

[25] R. J. Wilson. "An Eulerian Trail Through Königsberg". In: *Journal of Graph Theory* 10.3 (1986), pp. 265–275.

Current version of this topic finished on Dec. 12th, 2017.

Topic 4

The Chains of Andrei Andreevich Markov - I

Finite Markov Chains and Their Applications

He was too young to have been blighted by the cold world's corrupt finesse; his soul still blossomed out, and lighted at a friends word, a girl's caress.

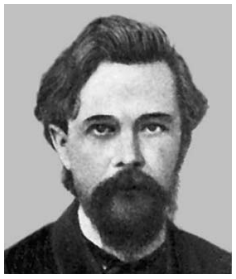
Alexander Pushkin, Eugene Onegin[1]

Keywords: *applied probability, stochastic processes, limiting behavior; applications: weather prediction, credit risk, Google's PageRank, voter migration, simulation of Markov chains*

4.1 An Invitation

4.1.1 The Law of Large Numbers and a Theological Debate

Andreii Andreevich Markov was born on June 14, 1856 in Rjasan, about 200 km in the south-east of Moscow. After finishing classical gymnasium he studied mechanics and mathematics at the University of St. Petersburg where he became a disciple of P. L. Chebyshev, one of the most influential and prolific Russian mathematicians of the 19th century. Among the wide spread research interests of Chebyshev, ranging from analysis and probability to the theory of numbers, there was the *Law of Large Numbers (LLN)* which finally attracted Markov's attention. A first version of this fundamental law has been formulated and proved by Jakob Bernoulli (1654–1705). In his analysis of games of chance he has shown that in a prolonged sequence of *independent* random trials, each having only two possible outcomes, success or failure, the relative frequency of observing *success* comes close to the theoretical success probability p. Stochastic independence, however, is a crucial condition in Bernoulli's derivation of



Andrei A. Markov
1856–1922

[1] Novel in verse published in 1833, cited from Hayes (2013). This was part of Markov's original experiments on the statistics of language.

the law, and that didn't change until Markov's contributions to the theory of the LLN. In the 19th century the concept of independence was not fully understood. This lack of comprehension resulted in the remarkable effect that the LLN got into the focus of a passionate theological debate about the existence of *free will* versus *predestination*. The debate was initiated by Pavel Nekrasov (1853–1924), a Russian mathematician with excellent relations to the Russian orthodox church. In a paper published in 1902 Nekrasov argued that voluntary acts are expressions of free will. And as such they are like independent events in probability theory, there are no causal links between these events. The LLN applies only to such events and this is, as he said, supported by data collected in social sciences like crime statistics.

Markov strongly objected to this interpretation of the LLN and in 1906 he addressed the problem to derive and prove a LLN for *dependent trials*. For this purpose he devised a simple stochastic process having two states only. Markov was able to show that as the process evolves over time the *average times* the system spends in either of these states approach a limit. As a sample application he performed a statistical analysis of the first 20 000 letters of Alexander Pushkin's (1799–1837) poem *Eugene Onegin*. The two states of his process (he used the term *chain*) were: a letter is a vowel (state 1), a letter is a consonant (state 2). Then he counted how often a vowel is followed by a vowel, a consonant, and similarly he counted transitions from consonant to vowel (consonant). In this way he formed a matrix of transition probabilities which became basis of subsequent analysis and the demonstration of a LLW for dependent trials.

So, this is quite remarkable, the first practical application of Markov chains was statistical linguistics, and in this field of science they are used even today.

4.1.2 Let's start with a definition

Let $X_0, X_1, X_2, \ldots$ denote a sequence of random variables, each taking its values in some finite set $\mathcal{S}$. The sequence $\{X_n, n \geq 0\}$ is called a *stochastic process* and $\mathcal{S}$ is its *state space*. The index n of X_n usually denotes time which we assume to be discrete. Thus there is some sort of clock with ticks at times $n = 1, 2, 3, \ldots$ and at these clock ticks the process may or may not change its state. Hence, if an observer finds that the random event $\{X_n = k\}$ occurred, we say the process X_n is in *state* $k \in \mathcal{S}$ at time n.

The behavior of the random process $\{X_n, n \geq 0\}$ can be described in various ways. One is to record values attained by X_n and determine the *joint distribution*, that is the probability:

$$P(X_0 = k_0, X_1 = k_1, \ldots, X_n = k_n).$$

Equivalently, we may be interested in the probability of finding X_n in a particular state k_n, *given the whole history* of the process $\{X_n, n \geq 0\}$. In other

words, we are interested in the *conditional probability*

$$P(X_n = k_n | \underbrace{X_0 = k_0, X_1 = k_1, \ldots, X_{n-1} = k_{n-1}}_{History\ \mathcal{H}_{n-1}}). \tag{4.1}$$

Generally, it is extremely difficult to determine probabilities like (4.1), because the dependence of X_n on its history $\mathcal{H}_{n-1}$ may be rather complicated. But there are situations that can be handled easily.

One is *independence* in which case

$$P(X_n = k|\mathcal{H}_{n-1}) = P(X_n = k).$$

An example is rolling a single dice where X_n denotes the value shown at the n-the experiment. Here the state space is $\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$, and whatever numbers have turned up in the first $n-1$ experiments, we always have (provided the dice is *a fair* one):

$$P(X_n = k|\mathcal{H}_{n-1}) = P(X_n = k) = \frac{1}{6} \quad \textit{for all } k \in \{1, 2, 3, 4, 5, 6\}.$$

The case of independence is very well understood and probability theory provides us we so marvelous tools like a *Law of Large Numbers*, a *Central Limit Theorem* etc. for independent stochastic sequences. Since independence makes everything much easier, it is not surprising that these fundamental laws have been discovered quite early, they are known since the 18th century.

But what, if there is no independence? When in 1906 A. A. Markov addressed the problem of proving a Law of Large Numbers for sequences of *dependent* random trials, he did so by assuming a very specific and simple type of dependence: he assumed that from the whole history $\mathcal{H}_{n-1}$ of X_n only the state occupied at time $n-1$ counts. In other words:

$$P(X_n = j|\mathcal{H}_{n-1}) = P(X_n = j|X_{n-1} = i), \quad \text{for every pair } i, j \in \mathcal{S}. \tag{4.2}$$

A typical example is the game *Monopoly*. Here X_n is the position of your token on the board after the n-th move. No matter how you came into position $X_{n-1} = i$ (say), rolling the dice determines how many steps your token moves ahead on the board. Where your token lands at the end of that move in position $X_n = j$ (say), this depends only on the position before the move and the outcome of throwing the dice [2].

Formula (4.2) is a formal statement of what is known as the *Markov property*, the random process $\{X_n, n \geq 0\}$ is called a *finite state Markov chain*. The basic structural parameters of a Markov chain (MC) are the *one-step transition probabilities*:

$$p_{ij} = P(X_n = j|X_{n-1} = i). \tag{4.3}$$

[2] You can find a nice analysis of *Monopoly* as a Markov Chain in the paper of Ash and Bishop (2003).

In a rather general setting these probabilities may depend on time n, i.e. the p_{ij} may vary with n as functions of time $p_{ij}(n)$. However, in this invitation (and in your thesis also), we shall assume that transition probabilities are *independent of time*. A MC having this property is called *time-homogeneous*.

The double indexing of these probabilities p_{ij} suggests to combine them in a *square matrix* $\mathbf{P}$. If the state space $\mathcal{S}$ has size $|\mathcal{S}| = N$, then $\mathbf{P}$ is a matrix of order $N \times N$:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{bmatrix}.$$

$\mathbf{P}$ is called the *transition matrix* of the MC $\{X_n, n \geq 0\}$. It has two rather obvious properties:

- The components are nonnegative: $p_{ij} \geq 0$ or all $i, j \in \mathcal{S}$, because these numbers are probabilities.
- The sum of each row equals 1, i.e., for each i we have $\sum_{j=1}^{N} p_{ij} = 1$. This is because the chain $\{X_n, n \geq 0\}$ cannot leave its state space.

Any square matrix having these characteristics is called a *stochastic matrix*. We will see soon that these properties have remarkable and far-reaching consequences.

Besides the transition matrix $\mathbf{P}$ we need another ingredient, an *initial distribution* . This specifies the probability of X_0 occupying a particular state at time zero, viz. $P(X_0 = i)$ for all states i. It will be very convenient to arrange these initial probabilities in a *vector* which we will almost exclusively use as a *row vector*: [3]

$$\boldsymbol{\pi}_0^t = [\pi_{01}, \pi_{02}, \ldots, \pi_{0N}], \quad \text{where } \pi_{0i} = P(X_0 = i) \tag{4.4}$$

Because $\boldsymbol{\pi}_0$ represents a probability distribution, we have

$$\sum_{i=1}^{N} \pi_{0i} = 1. \tag{4.5}$$

Any vector with nonnegative components which sum to one is called a *probability vector*.

Since matrix algebra will play a prominent role in the sequel, why not restate the basic properties of $\mathbf{P}$ and $\boldsymbol{\pi}_0$ in terms of matrix operations?

For this purpose we need to define a *one-vector* $\mathbf{1}$ of order $N \times 1$ as a vector

[3] In this invitation we adhere to the convention that any vector $\mathbf{a}$ is always interpreted as a *column vector*. If it happens (and it will happen) that we need $\mathbf{a}$ as a row vector, then we simply *transpose* it to $\mathbf{a}^t$.

consisting entirely of ones:

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Furthermore, for any matrix $\mathbf{A}$ or vector $\mathbf{a}$, we write $\mathbf{A} \geq \mathbf{0}$ and $\mathbf{a} \geq \mathbf{0}$, if *all* components of this matrix (this vector) are nonnegative. The symbol $\mathbf{0}$ represents the zero matrix or the zero vector.

Then the basic properties of $\mathbf{P}$ and $\boldsymbol{\pi}$ described above can be written compactly as:

$$\begin{aligned} \mathbf{P} &\geq \mathbf{0} && \text{nonnegativity} \\ \boldsymbol{\pi} &\geq \mathbf{0} && \\ \mathbf{P} \cdot \mathbf{1} &= \mathbf{1} && \text{row sums equal to 1} \qquad (4.6) \\ \boldsymbol{\pi}_0^t \cdot \mathbf{1} &= 1 && \text{this is (4.5).} \qquad (4.7) \end{aligned}$$

Observe that the left hand side of (4.6) is the usual matrix product of the transition matrix with the one-vector. Also, (4.7) is the matrix product of the *row vector* $\boldsymbol{\pi}_0^t$ and the one-vector (a column vector!). As these are of orders $1 \times N$ and $N \times 1$, the result is a matrix of order 1×1, i.e., the scalar value 1.

The specification of the initial distribution $\boldsymbol{\pi}_0$ and the transition matrix $\mathbf{P}$ completely determines the stochastic evolution of the MC $\{X_n, n \geq 0\}$. So, quite naturally the question arises:

Given $\boldsymbol{\pi}_0$ and $\mathbf{P}$, how can we calculate the distribution of the states X_n at some time n?

Let us denote this distribution by the probability vector:

$$\boldsymbol{\pi}_n^t = [\pi_{n1}, \pi_{n2}, \ldots, \pi_{nN}], \quad \text{where } \pi_{ni} = P(X_n = i). \qquad (4.8)$$

So, how to calculate $\boldsymbol{\pi}_n$?

It will turn out that $\boldsymbol{\pi}_n$ can be obtained by solving a simple *recurrence relation*. For the latter we can even find an explicit solution which, quite remarkably, is a *power law*!

4.1.3 Example 1: Will We Have a White Christmas This Year?

Every year at the beginning of December the question "Will we have a White Christmas this year?" pops up quite regularly in the weather shows of almost all TV-channels. Markov chain analysis may help us to shed some light on this really important question.

Rotondi (2010) approaches this forecasting problem by defining a two-state MC with state space

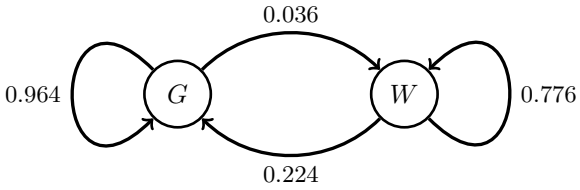
$$\mathcal{S} = \{G = \text{green day}, W = \text{white day}\}.$$

A green day is defined as a day with observed snow depth < 50 mm, whereas a white day must have snow depth ≥ 50 mm. Snow depth data are available from the *Global Historical Climatology Network (GHCN)* which collects data from weather stations all over the world. One such station is located in the Central Park in New York. Over several years for the period from December 17th to December 31st transitions[4] between the states W and G are counted and the observed relative frequencies are taken as statistical estimates of transition probabilities. This resulted in the transition matrix:

$$\mathbf{P} = \begin{matrix} \\ \text{G} \\ \text{W} \end{matrix} \begin{matrix} \begin{matrix} \text{G} & \text{W} \end{matrix} \\ \begin{bmatrix} 0.964 & 0.036 \\ 0.224 & 0.776 \end{bmatrix} \end{matrix}$$

Thus the probability p_{GG} that a green day is followed by a green day equals 0.964, so it's very likely that a green day follows a green day. Also, a green day is followed by a white day with probability $p_{GW} = 0.036$. In an analogous manner we interpret the second row of $\mathbf{P}$.

It will be very convenient to draw a diagram of the possible transitions in this chain:



Technically speaking, this is a *directed graph* with nodes corresponding to states and arcs (links) corresponding to possible transitions of the chain. Each arc $i \to j$ is assigned a *weight* which equals the transition probability p_{ij}. This graph is called the *transition graph* of the chain $\{X_n, n \geq 0\}$. No special knowledge in graph theory is required for this topic, but if you want to know more, you may consult the Invitation to Topic 2, *Shortest Paths in Networks*, which provides you with a rudimentary overview of the basic terminology from the theory of graphs.

Let us attack our weather forecasting problem *experimentally*. All we need is a computer, actually a pocket calculator suffices. Hard-nosed guys amonmg you can do the job with paper and pencil only.

Suppose that today is December 17th and this is a green day. This assumption specifies the following initial distribution:

$$\pi_{0G} = P(X_0 = G) = 1, \quad \pi_{0W} = P(X_0 = W) = 0 \implies \boldsymbol{\pi}_0^t = [1,\ 0]$$

[4]This period was chosen in order to minimize bias due to seasonality of weather patterns.

What is the probability that tomorrow, the day after tomorrow, etc. we will have a green or white day?

To determine these probabilities we have to invoke the *Law of Total Probability* : if there is an aribtrary event A and further events $B_1, B_2, \ldots, B_n$ which are *mutually disjoint* and whose union equals the sample space Ω, i.e.,

$$B_i \cap B_j = \varnothing \quad \text{for all } i \neq j, \quad \text{and} \quad B_1 \cup B_2 \cup \ldots \cup B_n = \Omega,$$

then the *total probability* of A is given by

$$P(A) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + \ldots + P(A|B_n)P(B_n). \tag{4.9}$$

In our case with $A = \{X_1 = G\}$ and two conditions $B_1 = \{X_0 = G\}$ and $B_2 = \{X_0 = W\}$ we get:

$$\begin{aligned} P(X_1 = G) &= P(X_1 = G|X_0 = G)P(X_0 = G) + P(X_1 = G|X_0 = W)P(X_0 = W) \\ &= \pi_{0G} \cdot p_{GG} + \pi_{0W} \cdot p_{WG} \\ &= 1 \cdot 0.964 + 0 \cdot 0.224 = 0.964 \end{aligned} \tag{A}$$

Similarly,

$$\begin{aligned} P(X_1 = W) &= P(X_1 = W|X_0 = G)P(X_0 = G) + P(X_1 = W|X_0 = W)P(X_0 = W) \\ &= \pi_{0G} \cdot p_{GW} + \pi_{0W} \cdot p_{WW} \\ &= 1 \cdot 0.036 + 0 \cdot 0.776 = 0.036 \end{aligned} \tag{B}$$

Not very spectacular, but looking closer at (A) and (B) you will find that both probabilities are *sums of products*, and whenever you encounter such a pattern, be sure, there's a matrix multiplication lurking behind. Indeed:

$$\boldsymbol{\pi}_1^t = [0.964,\ 0.036] = [1,\ 0] \begin{bmatrix} 0.964 & 0.036 \\ 0.224 & 0.776 \end{bmatrix}$$

In other words, we have found for our example:

$$\boldsymbol{\pi}_1^t = \boldsymbol{\pi}_0^t \cdot \mathbf{P} \tag{4.10}$$

So what about $\boldsymbol{\pi}_2^t$, $\boldsymbol{\pi}_3^t$, etc., the distribution of states in two, three days?

That's easy, the *Markov Property* (4.2) comes to our help. The latter implies that

$$\boldsymbol{\pi}_2^t = \boldsymbol{\pi}_1^t \cdot \mathbf{P},$$

but, by (4.10),

$$\boldsymbol{\pi}_2^t = (\boldsymbol{\pi}_0^t \mathbf{P}) \cdot \mathbf{P} = \boldsymbol{\pi}_0^t \cdot \mathbf{P}^2$$

Continuing in this manner we obtain:

$$\boldsymbol{\pi}_3^t = \boldsymbol{\pi}_2^t \cdot P = (\boldsymbol{\pi}_0^t \mathbf{P}^2) \cdot \mathbf{P} = \boldsymbol{\pi}_0^t \mathbf{P}^3$$

These easy-to-grasp steps when properly continued show us that there exists a fundamental *recurrence relation* having a very simple resolution as a *power law*:

$$\boldsymbol{\pi}_n^t = \boldsymbol{\pi}_{n-1}^t \mathbf{P} \implies \boldsymbol{\pi}_n^t = \boldsymbol{\pi}_0^t \mathbf{P}^n, \quad n = 1, 2, \ldots \tag{4.11}$$

Moreover, at no place in our calculations we have made use of the fact that $\mathbf{P}$ is only of order 2×2. Indeed, the law of total probability holds for any finite number of conditioning events. Hence it follows that the basic recurrence (4.11) holds generally for *all finite state Markov chains*.

Let's use now (4.11) to calculate the state probabilities for our example chain. Of course, normally we will not perform calculations by hand. It is much more convenient to use some computing environment which supports matrix calculations, e.g., `matlab` or its free clone `octave`. Alternatively, you may also use **R**. The latter is somewhat special, I will have to say more about this tool in Section 4 below.

Starting with a green day, our initial probability vector equals $\boldsymbol{\pi}_0^t = [1, 0]$, we obtain successively:

	day n	$P(X_n = G)$	$P(X_n = W)$
December 17th	0	1.0000	0.0000
	1	0.9640	0.0360
	2	0.9374	0.0626
	3	0.9176	0.0824
	4	0.9031	0.0969
	5	0.8923	0.1077
	6	0.8843	0.1157
Christmas Eve	7	0.8784	0.1216
	...		
New Year's Eve	14	0.8636	0.1364
	...		
January 16th	30	0.8615	0.1385
January 17th	31	0.8615	0.1385
	...		

Hm, that looks interesting.

Our calculations show: starting with a green day on December 17th, it is very likely that we won't have a White Christmas, in fact only with probability 12.2 % there will be snow on that day. Actually, the probability is about 88 % that December 24th is a green day again. Note however that this is not the probability of having 7 green days in a row. It just means that after 7 days we are again in state G.

But, more can be seen: there is a remarkable pattern, it seems that the state

probabilities *approach a limit*. Starting with a green day we expect *in the long run* a green day with probability 86 % and a white day with probability 14 %.

Having become curious through these observations we continue our experiment: this time we start with a white day, so $\boldsymbol{\pi}_0^t = [0,\ 1]$. Then repeating the procedure we obtain:

	day n	$P(X_n = G)$	$P(X_n = W)$
December 17th	0	0.0000	1.0000
	1	0.2240	0.7760
	2	0.3898	0.6102
	...		
Christmas Eve	7	0.7569	0.2431
	...		
New Year's Eve	14	0.8488	0.1512
	...		
January 16th	30	0.8614	0.1386
	...		
January 26th	40	0.8615	0.1385
January 27th	41	0.8615	0.1385
	...		

Now the probability of a white Christmas is 24 %. And interestingly, the long run distribution of green and white days is the same as before.

These are quite remarkable and unexpected observations. Let us summarize what we have found so far *experimentally*:

- It seems that the state distribution $\boldsymbol{\pi}_n$ approaches a *limit* as $n \to \infty$.
- Also, there is some indication that this limit is independent of the initial state distribution $\boldsymbol{\pi}_0$. We have found the same limit for the two initial distributions $[1, 0]$ and $[0, 1]$. We could have started also with another initial distribution of green and white days, for instance, we could have taken

  $$\boldsymbol{\pi}_0^t = [\ 0.5,\ 0.5\ ].$$

  Running through the recurrence we will again find that the state distribution settles in $[\ 0.8615,\ 0.1385\ ]$.
- It is not implausible that these observations must be somehow related to special properties of the transition matrix $\mathbf{P}$. Indeed, if we let the computer calculate some powers of $\mathbf{P}$, we find another strange pattern:

$$\mathbf{P}^{10} = \begin{bmatrix} 0.86836 & 0.13164 \\ 0.81912 & 0.18088 \end{bmatrix}, \quad \mathbf{P}^{20} = \begin{bmatrix} 0.86187 & 0.13813 \\ 0.85945 & 0.14055 \end{bmatrix}$$

$$\mathbf{P}^{40} = \begin{bmatrix} 0.86154 & 0.13846 \\ 0.86153 & 0.13847 \end{bmatrix}, \quad \mathbf{P}^{80} = \begin{bmatrix} 0.86154 & 0.13846 \\ 0.86154 & 0.13846 \end{bmatrix} \quad \ldots$$

Thus, the powers $\mathbf{P}^n$ themselves approach a limit, their rows are getting closer and closer to the conjectured limiting distribution of green and white days.

The observations we have made are typical of *regular chains*, a term of great importance in the theory of finite MCs. A MC is called *regular*, if from some n onward in the powers $\mathbf{P}^n$ there are no zeros. More formally:

$$\mathbf{P}^n > \mathbf{0} \quad \text{for sufficiently large } n.$$

A necessary (though not sufficient) condition for a chain to be regular can be identified by its transition graph: it must be possible to find a path of whatever length (= number of arcs comprising the path) between any pair of states. In terms of graph theory: the transition graph is *strongly connected*.

For regular chains there holds the following *fundamental theorem*:

- The powers $\mathbf{P}^n$ of the transition matrix approach a limiting matrix $\mathbf{A}$ as $n \to \infty$:

$$\lim_{n\to\infty} \mathbf{P}^n = \mathbf{A} \tag{4.12}$$

- Each row of $\mathbf{A}$ is the same vector $\boldsymbol{\alpha}^t$. This is the *limiting distribution* of states which is independent of the initial distribution $\boldsymbol{\pi}_0$ of the chain. In other words, whatever the initial distribution $\boldsymbol{\pi}_0$, the sequence $\boldsymbol{\pi}_n$ generated by the basic recurrence

$$\boldsymbol{\pi}_n^t = \boldsymbol{\pi}_{n-1}^t \mathbf{P}, \quad n = 1, 2, \ldots \tag{4.13}$$

  has the limit

$$\lim_{n\to\infty} \boldsymbol{\pi}_n^t = \boldsymbol{\alpha}^t \tag{4.14}$$

  Since the probabilities must sum to one, the vector $\boldsymbol{\alpha}$ must satisfy the sum condition

$$\boldsymbol{\alpha}^t \cdot \mathbf{1} = 1 \tag{4.15}$$

- The limiting probability vector $\boldsymbol{\alpha}$ is also a *stationary distribution* of the chain $\{X_n, n \geq 0\}$ in the sense that

$$\boldsymbol{\alpha}^t \mathbf{P} = \boldsymbol{\alpha}^t \tag{4.16}$$

  This follows directly from the existence of the limit (4.14) when applied to the recurrence (4.13). The term *stationary* simply says that if the chain starts with initial distribution $\boldsymbol{\alpha}$ the distribution of states after one, two, etc. steps will always be the same, it will never change. Indeed,

$$\boldsymbol{\alpha}^t \mathbf{P}^2 = (\boldsymbol{\alpha}^t \mathbf{P})\mathbf{P} = \boldsymbol{\alpha}^t \mathbf{P} = \boldsymbol{\alpha}^t,$$

  and similarly for any $n \geq 1$ we have $\boldsymbol{\alpha}^t \mathbf{P}^n = \boldsymbol{\alpha}^t$.

Returning to our snowfall example: this chain is certainly regular *ab initio* as $\mathbf{P} > \mathbf{0}$ and

$$\mathbf{A} = \lim_{n\to\infty} \mathbf{P} = \begin{bmatrix} 0.8615 & 0.1385 \\ 0.8615 & 0.1385 \end{bmatrix}, \quad \boldsymbol{\alpha}^t = [0.8615,\ 0.1385].$$

The way we have calculated the limiting distribution $\boldsymbol{\alpha}$ is known as *power method* because

$$\boldsymbol{\pi}_1^t = \boldsymbol{\pi}_0^t \mathbf{P}, \quad \boldsymbol{\pi}_2^t = \boldsymbol{\pi}_0^t \mathbf{P}^2, \quad \boldsymbol{\pi}_3^t = \boldsymbol{\pi}_0^t \mathbf{P}^3 \ldots$$

which is just a more explicit way of writing the the recurrence (4.13).

At the outset it is by no means clear that the sequence generated by the power method converges (and indeed, there are *non-regular* MCs where it does not!). Also, it may be that convergence is very slow and as a result a substantial amount of computational work may be necessary to come sufficiently close to the limit.

There are several alternatives to the power method. A quite efficient and elegant one is to start with the stationary equation (4.16) and the condition $\boldsymbol{\alpha}^t \cdot \mathbf{1} = 1$.

A close look at (4.16) reveals that it is actually a system of linear equations in the unknowns $\alpha_1, \alpha_2, \ldots, \alpha_N$, the components of $\boldsymbol{\alpha}$. It can be rewritten as:

$$\boldsymbol{\alpha}^t = \boldsymbol{\alpha}^t \mathbf{P} \implies \boldsymbol{\alpha}^t (\mathbf{I} - \mathbf{P}) = \mathbf{0}, \tag{4.17}$$

where $\mathbf{I}$ denotes the identity matrix. This system does not have a unique solution but that can be enforced by adding the sum condition (4.15) to (4.17).

Let's try this with our snowfall example. We have $\boldsymbol{\alpha}^t = [\alpha_G, \alpha_W]$ and

$$\mathbf{I} - \mathbf{P} = \begin{bmatrix} 0.036 & -0.036 \\ -0.224 & 0.224 \end{bmatrix},$$

therefore the matrix equation (4.17) and (4.15) can be written more explicitly as:

$$\begin{array}{rcrcll} 0.036\,\alpha_G & - & 0.224\,\alpha_W & = & 0 & \boldsymbol{\alpha}^t \times \text{ 1st column of } \mathbf{I}-\mathbf{P} \\ -0.036\,\alpha_G & + & 0.224\,\alpha_W & = & 0 & \boldsymbol{\alpha}^t \times \text{ 2nd column of } \mathbf{I}-\mathbf{P} \\ \alpha_G & + & \alpha_W & = & 1 & \end{array}$$

The first equation can be removed as it is essentially identical with the second equation. Solving the last two equations yields:

$$\alpha_G = \frac{0.224}{0.26} = 0.86154, \quad \alpha_w = \frac{0.036}{0.26} = 0.13846,$$

which conforms nicely to our experimental results.

Although this approach looks more attractive than the power method there are situations where the latter is really the method of choice. A remarkable example is the MC used in *Google's PageRank* algorithm which has several trillions of

states. In this case it is practically impossible to solve they system of linear equations (4.17) because it is too large.

So far some basic ideas to *regular Markov chains*. Much more can be said about these, I'll postpone this discussion to Section 2 were you will also find several really interesting applications among them Google's famous PageRank.

Not all MCs are regular, however. *Absorbing chains* are rather different, as our next example shows.

4.1.4 Example 2: Losing Your Money - Delinquency Of Loans

Lending money is always a risky business, as everybody knows. In accounting Markov chain models are often used as probability models for accounts receivable. Rating models like those of Standard & Poor's or Moody are well-known examples. Here we shall analyze a rather simple model discussed in Grimshaw and Alexander (2011). The basic idea of this model is: accounts receivable move through different *delinquency states* each month. For instance, an account in the state *current* (state C) this month will be in the state *current* next month, if a payment has been made by due date, and it will move to the state *delinquent* (state D), if no payment has been received. It may also happen that the account in state *current* is completely repaid, this is state R. An account in the state *delinquent* (D) may become a *loss* L or default, if the borrower fails to pay and there is no realistic hope that he will ever repay the loan.

A simple MC for this model thus has four states: C (*current*), D (*delinquent*), L (*loss or default*) and R (*repaid*).

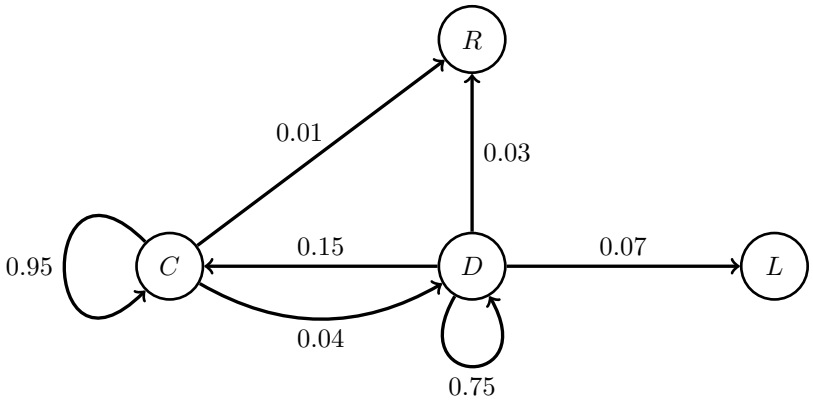
Suppose, following data for state transition probabilities are available:

$$\mathbf{P} = \begin{array}{c} \begin{array}{cccc} \text{C} & \text{D} & \text{L} & \text{R} \end{array} \\ \left[\begin{array}{cccc} 0.95 & 0.04 & 0 & 0.01 \\ 0.15 & 0.75 & 0.07 & 0.03 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \text{C} \\ \text{D} \\ \text{L} \\ \text{R} \end{array}$$

Observe that the rows corresponding to the states L and R are special, all components are zero except for $p_{LL} = p_{RR} = 1$. This means that whenever the chain enters either of those states, it gets trapped there, these states can never be left again. Accordingly, L and R are called *absorbing states*. Their special character comes out clearly when we draw the transition graph of this chain, as it is shown below. Observe that states R and L have only incoming arcs, no outgoing ones. For the other states C and D we find that they *communicate*. Between these states a joining path can be found in any direction. Thus our state space $\mathcal{S}$ naturally decomposes into two disjoint subsets:

$$\mathcal{S} = \mathcal{T} \cup \mathcal{A}, \quad \text{with } \mathcal{T} = \{C, D\}, \quad \mathcal{A} = \{L, R\}$$

The set $\mathcal{T}$ is called *transient* because sooner or later any borrower will be in an



absorbing state[5].

In Example 1 we started our analysis by experimentation, this time we will rely an *matrix algebra*. We will able to find out the form of $\mathbf{P}^n$ and much more by exploiting the special structure of $\mathbf{P}$. This structure is easy to see:

$$\mathbf{P} = \left[\begin{array}{cc|cc} 0.95 & 0.04 & 0 & 0.01 \\ 0.15 & 0.75 & 0.07 & 0.03 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} \mathbf{T} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{4.18}$$

Thus $\mathbf{P}$ can be divided into four blocks: a square matrix $\mathbf{T}$ governing the transition between transient states:

$$\mathbf{T} = \begin{bmatrix} 0.95 & 0.04 \\ 0.15 & 0.75 \end{bmatrix},$$

Furthermore there's a rectangular matrix $\mathbf{R}$ holding the probabilities of one-step transitions from any transient state into one of the two absorbing states:

$$\mathbf{R} = \begin{bmatrix} 0 & 0.01 \\ 0.07 & 0.03 \end{bmatrix}.$$

In the bottom row of $\mathbf{P}$ we have a 2×2 matrix of zeros $\mathbf{0}$ and a 2×2 identity matrix $\mathbf{I}$.

The *partition* (4.18) is called a 2×2 *block matrix*. The nice thing about such a matrix is, it can be multiplied with itself in much the same way as we do this with any 2×2 matrix having scalar components. We only have to take care of the fact that when multiplying sub-blocks the commutative law will not hold in general because these blocks are matrices.

So let's calculate $\mathbf{P}^2$. The standard multiplication scheme is:

$$\begin{array}{cc|cc} & & \mathbf{T} & \mathbf{R} \\ & & \mathbf{0} & \mathbf{I} \\ \hline \mathbf{T} & \mathbf{R} & \mathbf{T}\cdot\mathbf{T}+\mathbf{R}\cdot\mathbf{0} & \mathbf{T}\cdot\mathbf{R}+\mathbf{R}\cdot\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0}\cdot\mathbf{T}+\mathbf{I}\cdot\mathbf{0} & \mathbf{0}\cdot\mathbf{R}+\mathbf{I}\cdot\mathbf{I} \end{array} \implies \mathbf{P}^2 = \begin{bmatrix} \mathbf{T}^2 & \mathbf{TR}+\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

[5] *In the long run we are all dead.* (John Maynard Keynes, 1883-1946)

Similarly, we calculate $\mathbf{P}^3 = \mathbf{P}^2 \cdot \mathbf{P}$:

$$\mathbf{P}^3 = \begin{bmatrix} \mathbf{T}^3 & \mathbf{T}^2\mathbf{R} + \mathbf{T}\mathbf{R} + \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{T}^3 & (\mathbf{I} + \mathbf{T} + \mathbf{T}^2)\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

Can you see the *pattern*? Obviously $\mathbf{P}^n$ will look like

$$\mathbf{P}^n = \begin{bmatrix} \mathbf{T}^n & (\mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \ldots + \mathbf{T}^{n-1})\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

That's a very simple structure, except of the upper right corner. But again that expression reminds us on something. It's essentially a *geometric series*! We can find its sum in exactly the same way[6] as ordinary geometric series are summed, because the matrix $\mathbf{T}$ will commute with any of its powers $\mathbf{T}^k$.

Recall the classical summation formula for scalar geometric series

$$1 + a + a^2 + \ldots + a^{n-1} = \frac{1 - a^n}{1 - a}, \quad \text{provided } a \neq 1,$$

and in the limit:

$$\lim_{n\to\infty} (1 + a + a^2 + \ldots + a^{n-1}) = \frac{1}{1 - a}, \quad \text{provided } |a| < 1,$$

These formulas hold *verbatim* also for *matrix geometric series*:

$$\mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \ldots + \mathbf{T}^{n-1} = (\mathbf{I} - \mathbf{T}^n)(\mathbf{I} - \mathbf{T})^{-1}, \tag{4.19}$$

provided $\mathbf{I} - \mathbf{T}$ has an inverse. Fortunately, it can be shown that $\mathbf{T}^n$ converges to a zero matrix component-wise, i.e. $\lim_{n\to\infty} \mathbf{T}^n = \mathbf{0}$ which implies the existence of the inverse of $\mathbf{I} - \mathbf{T}$. This is the matrix analogue of the limiting relation $\lim_{n\to\infty} a^n = 0$ when $|a| < 1$. It is important to keep in mind that all these claims need a *proof*, see Section 2.

It follows that (4.19) becomes in the limit:

$$\lim_{n\to\infty} (\mathbf{I} + \mathbf{T} + \mathbf{T}^2 + \ldots + \mathbf{T}^{n-1}) = (\mathbf{I} - \mathbf{T})^{-1} \tag{4.20}$$

So putting things together we have found something really remarkable:

- The n-step transition matrix of an absorbing chain is given by:

$$\mathbf{P}^n = \begin{bmatrix} \mathbf{T}^n & (\mathbf{I} - \mathbf{T}^n)(\mathbf{I} - \mathbf{T})^{-1}\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{4.21}$$

- In the long run the behavior of an absorbing chain is governed by the limiting matrix:

$$\lim_{n\to\infty} \mathbf{P}^n = \begin{bmatrix} \mathbf{0} & (\mathbf{I} - \mathbf{T})^{-1}\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{4.22}$$

  because $\mathbf{T}^n \to \mathbf{0}$.

[6] You may consult any textbook on elementary mathematics.

In order to avoid our discussion becoming too academic, let's use our example data from above. In particular, we shall determine the one-year transition matrix $\mathbf{P}^{12}$.

Using any computing device supporting matrix calculations, you will find:

$$\mathbf{P}^{12} = \left[\begin{array}{cc|cc} 0.6751 & 0.1156 & 0.0777 & 0.1316 \\ 0.4335 & 0.0971 & 0.2994 & 0.1670 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

The result is interesting. From the first row it follows about 7.8 % of loans being in state C initially have defaulted by the end of the year or *even earlier*. Among those loans that were initially delinquent (row 2) the corresponding probability is almost 30 %.

The *long run behavior* is determined by (4.22):

$$\lim_{n\to\infty} \mathbf{P}^n = \left[\begin{array}{cc|cc} 0 & 0 & 0.4308 & 0.5692 \\ 0 & 0 & 0.5385 & 0.4615 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

This is remarkable again, the limiting matrix has *different rows*. In contrast to *regular* chains, for an absorbing chain the limiting distribution depends on the initial distribution. If we started with a good loan (state C), the initial distribution and its limit are

$$\boldsymbol{\pi}_0^t = [1, 0, 0, 0], \quad \lim_{n\to\infty} \boldsymbol{\pi}_0^t \mathbf{P}^n = [0,\ 0,\ 0.4308,\ 0.5692]$$

thus in the long run 43 % would be lost, 57 % repaid, whereas for delinquent loans these rates are 54 % and 46 %, respectively. On the other hand, if we start with a portfolio of 50 % good and 50 % delinquent loans,

$$\boldsymbol{\pi}_0^t = [0.5,\ 0.5,\ 0,\ 0], \quad \lim_{n\to\infty} \boldsymbol{\pi}_0^t \mathbf{P}^n = [0,\ 0,\ 0.4846,\ 0.5154],$$

losses would be about 49 %. That raises an interesting question: as loans can be traded, what is an *optimal loan portfolio*?

The inverse appearing in (4.22) is of special interest, we will denote it by

$$\mathbf{N} = (\mathbf{I} - \mathbf{T})^{-1}.$$

The matrix $\mathbf{N}$ is commonly known as the *fundamental matrix* of an absorbing chain because a lot of interesting quantities can be derived from it. In our example:

$$\mathbf{N} = (\mathbf{I} - \mathbf{T})^{-1} = \left[\begin{array}{cc} 38.4615 & 6.1538 \\ 23.0769 & 7.6923 \end{array}\right].$$

$\mathbf{N}$ is called fundamental because its entries n_{ij} are *expectations*! It can be shown that for *transient states* $i, j \in \mathcal{T}$:

$$n_{ij} = \text{mean number of times the chain is in state } j, \text{ given it started in state } i.$$

The row sums therefore equal the mean number of steps until absorption.

$$\mathbf{N} \cdot \mathbf{1} = \begin{bmatrix} 38.4615 & 6.1538 \\ 23.0769 & 7.6923 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 44.6153 \\ 30.7692 \end{bmatrix}$$

For example, consider a delinquent loan. It takes on average 31 months until this loan is either repaid or lost.

4.2 Where to go from here

The topic *Finite Markov Chains* is an extraordinary wide one, regarding the theory behind it but also regarding its applications. The latter should be in the focus of your thesis, but nevertheless, you should not forget about basic theory.

In principle, there are two routes (at least) you may pursue:

- Your thesis may become a well balanced mix of theory and application. If you prefer this route, then watch out for one, may be two really interesting real-world applications of MCs. Describe them in details, their pros and cons, the data used, discuss extensions and generalizations, statistical estimation of transition probabilities is also an issue. Develop the theory of MCs (see below) as far as it is necessary for your application(s).
- Computer simulation of MCs is another fascinating topic, a suggestion is presented below. But be warned: although this sounds easy, it is not. And: keep your fingers off a class of methods which is known as *Markov Chain Monte Carlo* as far as it is concerned with Bayesian statistics. This is really beyond the scope of your thesis.

4.2.1 Make up your mind - absorbing or regular chains?

In the *Invitation* to this topic I have sketched some important theoretical concepts, though exposition has been kept on a minimum theoretical level, simply because my main intention was to raise your interest in this topic. But now its time to expand on your knowledge about MCs and incorporate this into your work.

If you want to discuss *absorbing chains* and some of their applications:

- Explain the concept of a *fundamental matrix*.
- Show (elementary!) that the matrix $\mathbf{T}^n \to \mathbf{0}$.
- Show that the entries in the fundamental matrix equal the expected duration of stay in a transient state.
- Determine the expected time to absorption and also its variance.

If you put your emphasis on applications of *regular chains* :

- There's also a fundamental matrix for regular chains. Give a formula and discuss its derivation.
- Give examples of various interesting quantities which can be derived from the fundamental matrix, e.g. discuss *first passage times*, i.e., the number of steps required to reach a given state for the first time.
- There is a *Law of Large Numbers* for the average number of visits to a state and a *Central Limit Theorem* for the number of visits. You should give an account of these most important results, though you should *not prove* them. That's too difficult.

Finally, if you decide to concentrate on simulation of MCs, then your thesis still should not forget about theory. Thus basics of regular and absorbing chains should be discussed, may be also of *periodic chains*.

For the theoretical part the famous book Kemeny and Snell (1983) will be very helpful, in particular chapters 3 and 4. Please have a look at the *Annotated Bibliography*, Section 3.

As I already remarked, examples are a most important part of your thesis. Grabbing in the net you will find a tremendous number of applications. But please have a look at Langville and Hilgers (2006), the title of this paper is a program: the five greatest applications of Markov Chains. So make up your mind and decide what you want to work on.

There are some applications which I found particularly interesting. You are invited to have a look at them, may be you find these interesting too.

4.2.2 Google's PageRank Algorithm

If you call Google's main page and type: `Markov chains applications` you get almost instantly (about 0.4 secs) 1 220 000 hits. The speed is impressing, of course. But even more impressing is that the search results are *ranked* by *importance* and it is this ranking which is to a greater part responsible of Google's economic success. PageRank's philosophy is that a webpage is important if it is pointed too by other important pages. The crucial step is to measure *relative importance* of pages. Sergei Brin and Larry Page, in 1999 students at Stanford University and later founders of Google, had the ingenious idea (Brin et al., 1999) to accomplish this task by representing the hyperlink structure of the world wide web as a gigantic transition graph. Each node in this graph represents a web page, and if a page has a link to another page then there is an arc pointing to that page.

It is possible to construct a transition probability matrix of enormous size equal to the number of web pages online, it has several trillions of rows and columns. To bring order into the web Sergei Brin, Larry Page and coworkers suggested to calculate the vector $\boldsymbol{\alpha}$ of the limiting distribution of the corresponding MC. Once $\boldsymbol{\alpha}$ is known, pages are ranked according to the values in this vector. Pages with high limiting probabilities get a high rank, those with small values get low

ranks[7].

There are several mathematical hurdles. For instance, it is by no means clear that the resulting chain is regular. And indeed, it is not because there exist *dangling pages* which have no outgoing links. But by some ingenious tricks it can be made regular (as our snowfall example). So the limiting distribution exists and can be computed, theoretically. In practice, however, this is highly nontrivial because of size. For the calculation of $\boldsymbol{\alpha}$ the power method is used with a number of additional measures taken which are not disclosed to the public. Still, the amount of computational work to be performed *regularly* is gigantic. Actually Moler (2002) has called this *the world's largest matrix computation*. It should be remarked that Google's PageRank method is not the only approach to the information retrieval problem in the WWW. More information on this and related applications of regular chains can be found in Langville and Meyer (2005) and the textbook Langville and Meyer (2006).

4.2.3 Credit Ratings

In Example 2 above we have already touched the credit business, a simple absorbing chain was constructed to describe the changes in state of debt. On a higher level, e.g. when dealing with government bonds various international agencies collect data around the world and publish ratings. For instance, Standard and Poor's has its *RatingsDirect* (Vazza and Kraemer, 2015) which is published every year and contains a lot of interesting data. Most interesting is *rating data*. S & P's are using a scale of seven grades ranging from *triple A (AAA)* over *BBB, BB* and *B* to *CCC/C*. Triple A is best, of course, *CCC/C* is rather bad, but not the worst case that can happen. There are two more ratings: *Default (D)* and *Not Rated (NR)*. Default means that investors have lost their money (very likely), and an asset in state *NR* is usually also not a good thing. S & P's publish tables with transition rates of assets or issuers changing between various rating levels. These tables are in principle *transition matrices* of MCs with nine states, two of them (*Default* and *NR*) are absorbing. The tables also show different levels of aggregation: there are global one-year transition matrices, matrices for USA, for Europe, for financial institutions and insurance business. Also transition matrices over longer periods, 5 years and 10 years are published.

Moody has a similar system with 22 states (see e.g. Metz and Cantor (2007)), two of them again absorbing: *Default (D)* and *Withdrawn (WR)*.

I think that these reports are a rich source of data and valuable information. So, that's a perfect playing ground to apply absorbing Markov chain theory.

[7] Bad guys (pages) which try to betray Google's search engine get zero rank!

4.2.4 Generating Random Text, maybe Bullshit

Generating random text by *simulating Markov chains* is a fascinating and sometimes also really funny business. Practically all so-called *bullshit generators* you can find in the web are based on this idea in one or another way. For instance Donald Trump's public speeches inspired some people familiar with basic Markov chain theory to create the speech generator located at `http://trump.frost.works/`[8].

The first publication I am aware of that discusses Markov chain text generation is the seminal paper on the mathematical foundations of communication theory by Claude Shannon (1948) who argued that any source transmitting data gives rise to a Markov chain. He also gives nice demonstrations and examples of text sequences generated by a MC.

Basically, we need three ingredients for MC text generation.

1. A sufficiently large *text corpus* to estimate the transition matrix of a finite-state MC.

2. An algorithm to simulate the MC given its transition matrix.

As a text corpus one may take e.g. a collection of public speeches of a famous politician, a novel like Tolstoi's *War And Peace*, etc. The corpus has to be split into *tokens* which may be single letters or groups of consecutive letters including punctuation and white space, but more interesting are tokens which are whole words from the text. Just to give you a very small example[9]: Suppose the text corpus is the following tweet of D. Trump from 28 Jan 2014 (for simplicity all characters lower case):

```
snowing in texas and louisiana, record setting freezing
temperatures throughout the country and beyond.
global warming is an expensive hoax!
```

As tokens we may define for instance all 2-groups (*bigrams*) of consecutive letters. This may remind you on Markov's analysis of *Eugene Onegin*. In our sample corpus (· represents white space):

```
sn no ow wi in ng g· ·i in n· ·t te ex xa as ...
```

Take these as the states of the chain and perform now transition counts. For instance in our sample there are 5 occurrences of the token `in`, the chain visits five times this state. Four times this token changes to `ng`, and only once to `n·`. From the counts you can easily construct a transition matrix.

[8] For the sake of fairness, there are also Hillary Clinton speech generators, see e.g. `http://www.themolecularuniverse.com/HillarySpeech/`

[9] You may also have a look at Topic 11 - *Elementary Methods of Cryptology* to see it live in a cryptanalytic context.

Alternatively, you could also use *trigrams*, but note that then your transition matrix becomes very large. Anyway, for a text generator to produce something coming close to human language, it may be more fruitful the define words or groups of consecutive words as tokens.

Having estimated a transition probability matrix $\mathbf{P}$ its really easy to simulate chains. Fix an initial state and then generate a sequence $u_1, u_2, \ldots$ of *pseudo random numbers* having a uniform distribution on the interval $[0, 1]$. For instance, when the chain is in state k, take the k-th row of $\mathbf{P}$. Its *cumulative sums* split the unit interval $[0, 1]$ into N contiguous subintervals, N being the number of states. Just check into which interval u_k falls. If it falls into the i-th subinterval, the chain jumps into state i. Then repeat the procedure, this time with row i of $\mathbf{P}$.

You may implement this two-step process (estimation, simulation) in any programming language you like, `java, python`, etc. But, fortunately there are software packages making life easier, for instance R has the package `markovchain`. I'll have to say more about that package later.

Whatever way you do it, you should check the properties of the transition matrix estimated. Strange things may happen. Maybe the chain is regular, fine. Maybe it has absorbing states. Then sooner or later your simulated chain will get trapped in one of these absorbing states and your text generator keeps on producing something like .....

But it may also happen that the chain gets trapped in a *subset* of states which results in a marked deprivation of the text generated. Also, it is possible that the chain exhibits *ultimately periodic behavior*. This is very interesting from a mathematical point of view, but not so really welcome for a text generator, when from some time onward it keeps printing an endless string of sort `bla bla bla ....`

4.2.5 Other Applications

Here are a few more interesting applications reviewed in telegraphic style:

- *Brand Switching Models*, also known as or related to so-called *brand choice models*. The behavior of consumers having the choice between different brands of some commodity can be modeled as a regular Markov chain. An analysis of these chains allows interesting statements about consumer loyalty and other aspects important from a marketing point of view. The paper of Colombo and Morrison (1989) is an easily accessible starting point.
- *Voter migration.* In western democracies there are strong tendencies that voters no longer stick to a particular party but tend to change to alternative political competitors. Impressive amounts of data are available today. It is not a new idea but pretty challenging to model voter migration as a MC. See the interesting websites maintained by Baxter (n.d.) or the *SORA Institute for Social Research and Consulting*, `www.sora.at`.

- *Production management.* Another application of absorbing Markov chain theory is modeling the flow of material through a production system. The stochastic character of this type of models is usually uncertainty due to possible reprocessing of parts because of insufficient quality as well as scrapping. The paper of Pillai and Chandrasekharan (2008) is exemplary in this context.

4.3 An Annotated Bibliography

A beautiful paper to begin with is Hayes (2013). The author presents a gentle introduction into the basics of Markov chains, there's also a *weather example* and a discussion of Markov's statistical analysis of Alexander Pushkin's *Eugene Onegin*. In addition it has a nice account of Markov's life and his work on Markov chains. Interesting details about Markov are revealed in the papers Langville and Hilgers (2006) and Basharin, Langville, and Naumov (2004). The former discusses also five great applications of MCs, among them Markov's original linguistic analyses and the PageRank algorithm of Google.

Markov chains are covered in practically all serious textbooks on probability, random processes and stochastic modeling. Unfortunately many of these books are not easily accessible to beginners and difficult to read. A major reason is the emphasis on Markov chains with infinite state spaces. But the mathematical theory for the latter is rather intricate. The standard textbook on finite chains is certainly Kemeny and Snell (1983). It is very well suited for beginners and elaborates clearly the matrix algebraic aspects of the subject. Unfortunately, the book is rather old and the notation used is somewhat nonstandard today. More demanding but still strongly recommended is Karlin and Pinsky (2011), in particular chapters 3 and 4. There you can also a lot of interesting applications carefully presented and worked out.

Caveat. In the process of preparing this topic I have read quite a number of recent publications on Markov chains and I noticed that it becomes more and more fashionable to construct transition probability matrices as *column-sum stochastic*. Where we have defined p_{ij} as the probability of a jump from state i to state j some authors do it the other way round, so p_{ij} now becomes the probability of a jump from j to i. Obviously, this is done to achieve some notational simplification. That's not worth it. I consider this bad style because it breaks with the tradition of all classical and serious texts on Markov chains.

4.4 A note on software

Applications of MCs including simulation is greatly facilitated by several software tools. The R `markovchain` package developed by Spedicato et al. (2015) is a rather comprehensive collection of software routines to handle finite MCs. Among standards like the calculation of fundamental matrices, it offers diagnostics for a chain to be regular, etc. There is also the possibility to draw

transition graphs, although you should not expect too much, unless the chain has a sufficiently small state space. Quite interesting for you are routines for statistical estimation of transition matrices and, last but not least, there is also a routine to simulate chains.

4.5 References

[1] R. B. Ash and R. L. Bishop. "Monopoly as a Markov Process". In: (2003). URL: www.math.uiuc.edu/~bishop/monopoly.pdf.

[2] G. P. Basharin, A. N. Langville, and V. A. Naumov. "The Life and Work of A. A. Markov". In: *Linear Algebra and its Applications* 386 (2004), pp. 3–26.

[3] Martin Baxter. *Electoral Calculus*. URL: http://www.electoralcalculus.co.uk.

[4] S. Brin et al. "The PageRank Citation Ranking: Bringing Order in the Web". In: *Technical Report 1999-0120, Computer Science Department. Stanford University* (1999).

[5] R. A. Colombo and D. G. Morrison. "A Brand Switching Model with Implications for Marketing Strategies". In: *Marketing Science* 8.1 (1989), pp. 89–99.

[6] S. D. Grimshaw and W. P. Alexander. "Markov chain models for delinquency: Transition matrix estimation and forecasting". In: *Applied Stochastic Models in Business and Industry* 27.3 (2011), pp. 267–279.

[7] Brian Hayes. "First Links in the Markov Chain". In: *American Scientist* 101 (2013), pp. 92–97.

[8] Samuel Karlin and Mark A. Pinsky. *An Introduction to Stochastic Modeling*. Academic Press, 2011.

[9] J. G. Kemeny and L. Snell. *Finite Markov Chains*. Springer, 1983.

[10] A. N. Langville and P. von Hilgers. *The Five greatest Applications of Markov Chains*. 2006. URL: http://langvillea.people.cofc.edu/MCapps7.pdf.

[11] A. N. Langville and Carl D. Meyer. "A Survey of Eigenvector Methods for Web Information Retrieval". In: *SIAM Review* 47.1 (2005), pp. 135–161.

[12] A. N. Langville and Carl D. Meyer. *Google's PageRank and Beyond. The Science of Search Engine Rankings*. Princeton University Press, 2006.

[13] Albert Metz and Richard Cantor. *Introducing Moody's Credit Transition Model*. 2007. URL: http://www.moodysanalytics.com/~/media/Brochures/Credit-Research-Risk-Measurement/Quantative-Insight/Credit-Transition-Model/Introductory-Article-Credit-Transition-Model.pdf.

[14] Cleve Moler. *The World's Largest Matrix Computation*. 2002. URL: https://www.mathworks.com/company/newsletters/articles/the-world-s-largest-matrix-computation.html.

[15] V. M. Pillai and M. P. Chandrasekharan. "An absorbing Markov chain model for production systems with rework and scrapping". In: *Computers and Industrial Engineering* 55 (2008), pp. 695–706.

[16] Michael A. Rotondi. "To Ski or Not to Ski: Estimating Transition Matrices to Predict Tomorrow's Snowfall Using Real Data". In: *Journal of Statistical Education* 18.3 (2010).

[17] Claude E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[18] G. A. Spedicato et al. *The markovchain Package: A Package for Easily Handling Discrete Markov Chains in R*. 2015. URL: https://cran.r-project.org/web/packages/markovchain/vignettes/an_introduction_to_markovchain_package.pdf.

[19] Diane Vazza and Nick W. Kraemer. *2014 Annual Global Corporate Default Study And Rating Transitions*. 2015. URL: https://www.nact.org/resources/2014_SP_Global_Corporate_Default_Study.pdf.

Topic 5

The Chains of Andrei Andreevich Markov - II

Finite Markov Chains and Matrix Theory

The theory of finite homogeneous Markov chains provides one of the most beautiful and elegant applications of the theory of matrices.

Carl D. Meyer, 1975

Keywords: *probability theory, stochastic processes, matrix algebra*

5.1 An Inivitation

This topic is under development and has not been finished yet.

Actually, when preparing Topic 4, *Finite Markov Chains and Their Applications*, I realized that it would be a good idea to make an own topic entirely devoted to matrix theory and applications. There is nothing to add to Carl D. Meyer's quote above.

This topic will cover presumably the following points:

- Nonnegative matrices and the *Perron-Frobenius Theorem*.
- Structural properties of a Markov transition matrix, reducibility.
- Convergence of transition matrices, summability.
- Spectral decomposition and *Sylvesters Formula*.
- Convergence of the *power method* for regular chains.
- The fundamental matrix as a generalized inverse.
- Stochastic complements and uncoupling Markov chains.
- . . . and may be more to come.

5.2 An Annotated Bibliography

Here are two interesting papers due to C. D. Meyer which strongly motivate me to work out this topic, Meyer (1975) and Meyer (1989).

5.3 References

[1] Carl D. Meyer. "Stochastic Complementation, Uncoupling Markov Chains, and the Theory of Nearly Reducible Systems". In: *SIAM Review* 31.2 (1989), pp. 240–272.

[2] Carl D. Meyer. "The Role of the Group Generalized Inverse in the Theory of Finite Markov Chains". In: *SIAM Review* 17.3 (1975), pp. 443–464.

Topic 6

Benford's Law

The Law of Anomalous Numbers is thus
a general probability law of widespread application.

Frank Benford, 1938

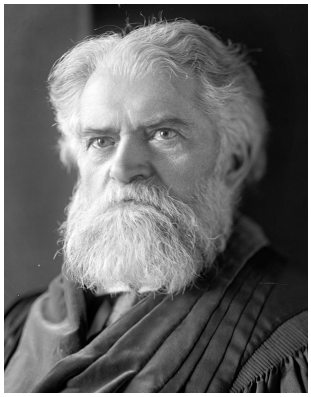
Keywords: Mathematical and statistical forensics,
experimental statistics, probabilistic number theory

6.1 An Invitation

6.1.1 Simon Newcomb and the First Digit Law

A distinguished applied mathematician was extremely successful in bets that a number chosen at random in the *Farmer's Almanach*, the *US Census Report* or a similar compendium would have the first significant digit less than 5. It is reported by Feller (1971) that this man won almost 70 % of his bets.

How is this possible? Normally the numbers we use in everyday live are expressed in decimal system so that the digits making up a number are integers in the range $0, 1, 2, \ldots 9$ except for the first or leading digit which by convention and convenience is never zero. If we look at a collection of numbers like a table of physical constants, atomic weights, populations counts of cities, distances of galaxies from the earth, numbers in annual reports of companies, *Farmers Almanach*, etc., we intuitively expect that all digits in these numbers should occur with equal frequency. Thus the leading digit, say D_1, should be $1, 2, \ldots, 9$ with frequency close to $1/9$, and the second digit D_2, the third D_3, etc. should take their values with equal frequencies $1/10$.



Simon Newcomb
(1835-1909)

Interestingly, very often this is not true, at least for the examples I have just mentioned and for may other examples as well. In 1881 the astronomer Simon Newcomb (1881) published a short note in the *American Journal of Mathematics* which he opened as follows:

> *That the ten digits do not occur with equal frequency must be evident to anyone making use of logarithm tables, and noticing how much faster the first pages wear out than the last ones.*

He argued that numbers whose first digit is small are more likely to be used than numbers with first digit being greater than 4 or 5, say. In particular, he stated:

> *The law of probability of the occurrence of numbers is such that all mantissae of their logarithms are equally probable.*

Newcomb's discovery did not have much resonance and apparently was forgotten until in 1938 Frank Benford, at that time physicist at General Electric, rediscovered Newcomb's result which since then is commonly known as *Benford's Law*[1] . Benford (1938) collected lots of data from various areas so diverse as numbers of inhabitants of towns, physical measurements, the *Farmer's Almanac*, atomic weights, voltages of X-ray tubes, results from sports leagues, powers and square roots of natural numbers, etc. For most of these data he found that the frequency (we are tempted to say, the probability) of the first digit D_1 is very close to the logarithmic law:

$$P(D_1 = d) = \log\left(1 + \frac{1}{d}\right), \quad d = 1, 2, \ldots, 9 \tag{6.1}$$

Here log means (as always in this introduction) logarithm to base 10. Formula (6.1) is commonly known as *Benford's First Digit Law*. We shall call it also the *weak version* of *Benford's Law*.

By simple calculation we obtain:

d	1	2	3	4	5	6	7	8	9
$P(D_1 = d)$	0.301	0.176	0.125	0.097	0.079	0.067	0.058	0.051	0.046

Table 6.1: Benford's Law - first digit probabilities

A bar plot of these data is displayed in Figure 1 below. It explains why the applied mathematician mentioned above had a chance of about 69.9 % to win his bets. Not so bad.

Benford called numbers following the logarithmic law *anomalous numbers* and observed that the fit to this law was even better when we combined data of very different sources into a single large sample. That's quite remarkable too because these data have very different units of measurement!

Mark Nigrini (1992) finished his PhD thesis on Benford's Law and suggested to use it as an auditing and accounting tool to detect anomalies in company data. Indeed, he found that most accounting data very closely follow Benford's

[1] Benford's article suffered a much better fate than Newcomb's paper, possibly in part because it immediately preceded a physics arcticle by Hans Bethe et al. on the multiple scattering of electrons (Miller, 2015).

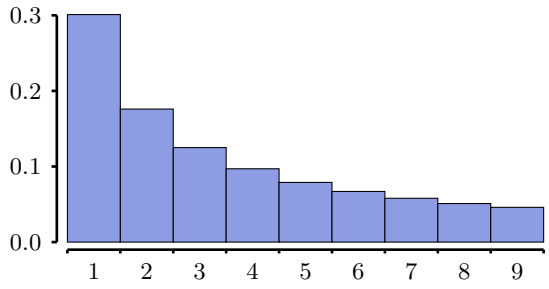


Figure 6.1: Benford's Law - first digit probabilities

Law. However, in case of accounting fraud this is quite often not the case. This significant deviation of overserved data from Benford's Law may be an indication that data have been manipulated, may be by fraudulent intent.

Nigrini's work initiated a new discipline nowadays known as *forensic statistics* or *analytic forensics*. Very quickly judicial authorities became aware of these new ideas. The *Wall Street Journal (July 10, 1995)* reported that the chief financial investigator for the district attorney's office in Brooklyn, N. Y., Mr. R. Burton *used Nigrini's program to analyze 784 checks issued by seven companies and found that check amounts on 103 checks didn't conform to expected patterns. "Bingo, that means fraud," says Mr. Burton. The district attorney has since caught the culprits, some bookkeepers and payroll clerks and is charging them with theft.* In particular Mr. Burton obtained the frequencies of first digits given in Table 2. Look how different these frequencies are from those predicted by Benford's Law. It seems that faking data in an intelligent way is not so easy and, probably Mafia people should learn more mathematics!

d	1	2	3	4	5	6	7	8	9
$f(d)$	0.000	0.019	0.000	0.097	0.612	0.233	0.01	0.029	0.000

Table 6.2: First digit frequencies of fraudulent data (Theodore P. Hill, 1999)

You might have got the impression that all numeric data follow Benford's Law, but this is not true. Here are a few examples of data which are not *Benford*: ZIP-codes, prices of products like € 1.99 often used in supermarkets for psychological reasons, telefone numbers, numbers drawn in lotteries, etc. Other examples are data sampled from a normal distribution, data generated by a random walk process, data having a narrow range of possible values like body-height etc.

Why are many data sets *Benford* and others are not? It has been argued that this may be a property inherent to our decimal number system but it turned out that this argument does not hold. Benford's Law can be observed also for binary, octal, hexagesimal numbers.

It has also been argued that there may be some *universal law* behind our data, probably comparable to one of the most fundamental laws in probability, the

Central Limit Theorem. It was not before 1995 when the pioneering works of Theodore Hill (see the annotated bibliography in Section 3 below) shed a bright light on this mysterious law.

6.1.2 The significand function

We are going now to work out some basic mathematical ideas which not only prove to be very useful but also allow us to state a much stronger version of Benford's Law.

In his 1938 paper Benford evaluated 20 229 data entries from various sources and it must have been a rather fatiguing process to count the digits of so many numbers. Of course, today we will do that not by hand but use the computer. But how can we persuade the computer to return the first or second or third digit of a given number? This requires some technique.

The *significand function* providing the basis of these techniques plays a fundamental role in the context of Benford's Law. Let $x > 0$ be a real number, then the significand function $S(x)$ is defined as

$$S(x) = t, \qquad t \in [1, 10) \tag{6.2}$$

where t is the unique number such that $x = t \cdot 10^\sigma$ for some necessarily unique $\sigma \in \mathbb{Z}$. The exponent σ is called the *scale* of x. For convenience we define $S(x) = 0$ when $x = 0$. Since we are interested in the digits of x, its sign will never play a role.

For example, for $x = 2356.88 = 2.35688 \cdot 10^3$ we have $S(x) = 2.35688$ with scale $\sigma = 3$.

How do we get the scale? For this purpose we need a most important *integer function*, the *floor* $\lfloor x \rfloor$, which is defined as the integer nearest to x when x is *rounded down*. Thus

$$\lfloor 4.81 \rfloor = 4, \quad \lfloor -2.4 \rfloor = -3, \quad \text{etc.}$$

The scale σ of a nonnegative number x is simply $\lfloor \log x \rfloor$. Indeed, $\log 2356.88 = 3.3723$, thus $\sigma = \lfloor 3.3723 \rfloor = 3$.

Note that σ is not very interesting for our purposes as we are primarily interested in the values of the digits and not in the position of the decimal point. Thus it will be convenient to strip off the integer part of a number x, which yields the *fractional part* of x, denoted by $\langle x \rangle$:

$$\langle x \rangle = x - \lfloor x \rfloor$$

The unique representation of $x = S(x) \cdot 10^\sigma$ implies $S(x) = x \cdot 10^{-\sigma}$. But for $x > 0$ it holds that $x = 10^{\log x}$ therefore we get the following explicit representation of the significand function valid for all $x \neq 0$:

$$S(x) = 10^{\log |x| - \lfloor \log |x| \rfloor} = 10^{\langle \log |x| \rangle} \tag{6.3}$$

The function $S(x)$ gives us direct access to the digits making up a number x.

Let $D_m(x) := D_m, m \in \mathbb{N}$, be the m-th significant digit of x when counted from left. Since we agree that a number will never have leading zeroes, clearly:

$$D_1 \in \{1, 2, \ldots, 9\}, \quad D_m \in \{0, 1, \ldots, 9\} \quad \text{for } m > 1.$$

Also, let

$$\begin{aligned} S(x) &= D_1 + D_2 \cdot 10^{-1} + D_3 \cdot 10^{-2} + \ldots \\ &= \sum_{m \in \mathbb{N}} D_m \cdot 10^{1-m} \end{aligned} \tag{6.4}$$

be the decimal expansion of the significand function.

For example, $S(2356.88) = 2.35688$, thus its significant digits are:

$$D_1 = 2,\ D_2 = 3,\ D_3 = 5,\ D_4 = 6,\ D_5 = 8,\ D_6 = 8,$$

and, of course by (6.4):

$$S(2356.88) = 2 + 3 \cdot 10^{-1} + 5 \cdot 10^{-2} + 5 \cdot 10^{-3} + 6 \cdot 10^{-4} + 8 \cdot 10^{-5} + 8 \cdot 10^{-6}.$$

Cleverly using the floor function we get the digits D_m easily from the significand function:

$$D_m = \lfloor 10^{m-1} S(x) \rfloor - 10 \lfloor 10^{m-2} S(x) \rfloor, \quad \text{for all } m \in \mathbb{N}. \tag{6.5}$$

In particular, the leading digit is given by

$$D_1 = \lfloor S(x) \rfloor - 10 \lfloor 10^{-1} S(x) \rfloor.$$

For instance, when $x = 2356.88$, then $S(x) = 2.35688$ and

$$D_2 = \lfloor 10 \cdot 2.35688 \rfloor - 10 \lfloor 10^0 \cdot 2.35688 \rfloor = \lfloor 23.5688 \rfloor - 10 \lfloor 2.35688 \rfloor = 3, \text{ etc.}$$

6.1.3 Benford's Law and the uniform distribution

Now we put some randomness into the story: let X be a random variable and $S(X)$ its significand function. We *define*: X satisfies *Benford's Law (strong version)*, if $S(X)$ has a *logarithmic distribution*:

$$P(S(X) \leq t) = \log t, \quad t \in [1, 10) \tag{6.6}$$

Note that $S(X)$ appears to be a continuous random variable. We shall see shortly that (6.6) implies Benford's First Digit Law but the converse is not true. It is easy to see that the logarithmic law (6.6) holds *if and only if* the logarithm of $S(X)$ has a *continuous uniform distribution* on $[0, 1]$. Apply in the equation above the substitition $\log t = s$. Then $t = 10^s$ and:

$$P(S(X) \leq 10^s) = s \qquad s \in [0, 1).$$

Using the explicit expression (6.3) for the significand we obtain

$$P(10^{\langle \log |x| \rangle} \leq 10^s) = s \qquad s \in [0,1).$$

Upon taking logs we get

$$P(\log\langle |X| \rangle \leq s) = P(\log S(X) \leq s) = s, \qquad s \in [0,1). \tag{6.7}$$

Thus we realize that $\log S(x) \sim \mathcal{U}(0,1)$, a continuous uniform distribution on the interval $[0,1]$.

These observations tell us that we will observe Benford's law, if the logs of the significands of observed data are close to a uniform distribution *et vice versa*.

Note, that (6.7) gives us a simple way to *generate* pseudo random numbers that follow Benford's Law. Just generate a random number $U \sim \mathcal{U}(0,1)$ and form $Z = 10^U$, then Z must be *Benford*.

6.1.4 The general digit law

The logarithmic distribution (6.6) has lots of information to offer. For brevity let $S(X) := S$ and recall that the significand S has a decimal expansion

$$S = D_1 + D_2 \cdot 10^{-1} + D_3 \cdot 10^{-2} + \ldots,$$

moreover, S is a *continuous* random variable, so $P(S \leq t) = P(S < t)$ because $P(S = t) = 0$.

From (6.6) we can easily *derive* not only Benford's First Digit Law, but also the distribution of the second digit D_2, of the third digit D_3, etc.

Let's begin with D_1 and consider the event $\{D_1 = d_1\}$. Some reflection shows that it is equivalent to the event

$$\{D_1 = 1\} \equiv \{d_1 \leq S < d_1 + 1\}$$

Thus

$$\begin{aligned} P(D_1 = d_1) &= P(d_1 \leq S < d_1 + 1) = P(S \leq d_1 + 1) - P(S \leq d_1) \\ &= \log(d_1 + 1) - \log(d_1) = \log\left(\frac{d_1 + 1}{d_1}\right) \\ &= \log\left(1 + \frac{1}{d_1}\right), \end{aligned}$$

which is Benford Law for the first significant digit. But without any problems we can get more. For the *joint distribution* of the first two digits D_1 and D_2 we obtain:

$$\begin{aligned} P(D_1 = d_1, D_2 = d_2) &= P(d_1 + d_1 10^{-1} \leq S < d_1 + (d_2 + 1) \cdot 10^{-1}) \\ &= \log\left(1 + \frac{1}{10d_1 + d_2}\right) \end{aligned} \tag{6.8}$$

You will have no difficulties to fill in the details. Of course, $1 \leq D_1 \leq 9, 0 \leq D_2 \leq 9$.

Example. If a number is randomly selected from a data set following Benford's Law (strong version), then the probability that it starts with $50\ldots$ equals

$$P(D_1 = 5, D_2 = 0) = \log\left(1 + \frac{1}{50}\right) \doteq 0.0086$$

To obtain the *marginal distribution* of D_2 we have to sum (6.8) over all possible values of d_1:

$$P(D_2 = d_2) = \sum_{k=1}^{9} \log\left(1 + \frac{1}{10k + d_2}\right) \tag{6.9}$$

A trite caclculation yields:

d	0	1	2	3	4	5	6	7	8	9
$P(D_2 = d)$	0.120	0.114	0.109	0.104	0.100	0.097	0.093	0.090	0.088	0.085

Table 6.3: Benford's Law - second digit probabilities

A quite instructive barplot of the disgtributions of the first and second digits is displayed in Figure 6.2. You can see, that the distribution of D_2 is already rather close to a discrete uniform distribution, and this pattern prevails when we consider the marginal distributions of D_m for $m > 2$. Continuing the arguments

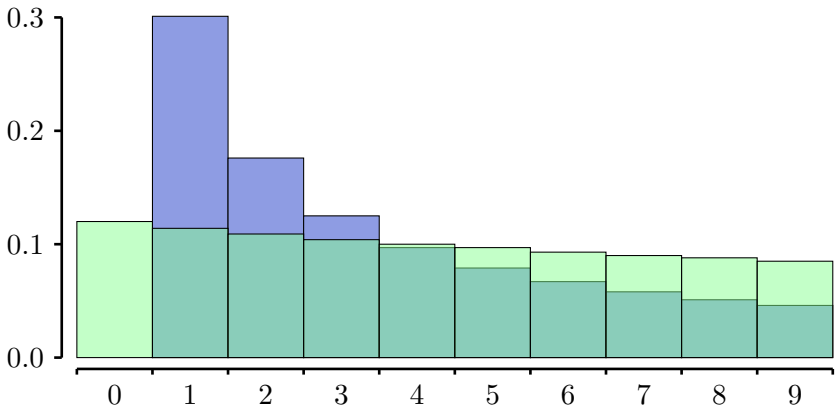


Figure 6.2: Benford's Law - first and second digit probabilities

outlined above it is not difficult to find the joint distribution of the first k digits:

$$\begin{aligned} P(D_1 = d_1, &D_2 = d_2, \ldots, D_k = d_k) = \\ &= \log\left(1 + \frac{1}{10^{k-1}d_1 + 10^{k-2}d_2 + \ldots + d_k}\right) \end{aligned} \tag{6.10}$$

Formula (6.10) is also known as *Benford's Law for the first k digits.*

6.1.5 Testing the Hypothesis

Benford or not Benford?

It is a question of utmost practical importance to find out whether a given data set conforms to Benford's Law. This question is a *statistical decision problem* and I will discuss very briefly how this problem can be addressed methodologically. I decided to include this in our *Invitation* for two reasons:

- A majority of publications related somehow to Benford's Law is empirical in nature.
- Many of these papers are relatively poor methodologically.

At the very beginning it is important to announce some *bad news*: no finite data set can be *exactly Benford* in the sense of (6.10). The reason is that the Benford probabilities (6.10) of sets of k given significant digits become arbitrarily small as $k \to \infty$, and no discrete probability distribution with finitely many points of support can take arbitrarily small positive values.

Despite of this drop of bitterness it is still legitimate to ask: how do we measure *close conformance* to Benford's law or a *significant deviation* from it? In statistics, this question runs commonly under the headline *goodness-of-fit*. We have a *null hypothesis*:

$$H_0 : \text{data conform to Benford's Law}$$

and the *alternative hypothesis*:

$$H_1 : \text{data do not conform to Benford's Law}$$

Given a significance level α, we want to decide whether for a given data set hypothesis H_0 has to be rejected or not. Or stated equivalently: *do our data show a statistically significant deviation from Benford's Law?*.

A rather traditional route of attack is this: use (6.10), usually for small values of k, and compare the predicted probabilities with empirical frequencies observed in sample data.

Virtually all empirical work on Benford's Law pursues this way and solves a *discrete goodness-of-fit problem*.

Separate testing of single digits

The idea is very simple and should be familiar to you from elementary statistics courses: suppose we want to find out whether there is a significant deviation between the observed frequency f_d of the event $\{D_1 = d\}$ and the the corresponding probability which should be $p_d = P(D_1 = d) = \log(1 + 1/d)$, as we know from (6.1). Testing conformity to Benford's First Digit Law is done for each the nine possible values d of D_1 separately:

$$\begin{aligned} H_0 &: \; p_d = \log(1 + 1/d), \\ H_1 &: \; p_d \neq \log(1 + 1/d) \end{aligned}$$

The test statistics are

$$T_d = \frac{|f_d - p_d|}{\sqrt{p_d(1-p_d)}}\sqrt{n}, \quad d = 1, 2, \ldots, 9$$

For large sample size n the statistics T_d are approximately standard normal, the corresponding p-values equal $P(|T_d| > |t_d|)$, where t_d is is the observed sample value of T_d.

The results of these tests have to be interpreted with care: Suppose that testing the nine hypotheses leads to a rejection of only the hypothesis for $d = 1$. Is this sufficient to conclude that our data are not *Benford*?

No, not at all. The point is, that performing these tests *simultaneously* affects the chosen level of significance and in turn the probability of type-II error. So this procedure should only be used in an explorative analysis of your data.

Distance-based tests

Separate testing of single digits is easy but because of lacking power this approach is not very reliable.

An alternative is testing based on the vectors

$$\mathbf{p} = [p_1, p_2, \ldots, p_9] \quad \text{and} \quad \mathbf{f} = [f_1, f_2, \ldots, f_9],$$

where the p_d are calculated according to (6.1), of course.

We are tempted to say that the observed frequencies $\mathbf{f}$ are *close* to the probability distribution under H_0 represented by $\mathbf{p}$, if some appropriate measure of deviation has a small value, otherwise we would reject H_0. To be concrete, our testinging problem is now more specifically:

H_0 : D_1 has distribution $\mathbf{p}$ and therefore is *Benford*

H_1 : D_1 has another distribution and therefore is not *Benford*

Several distance measure are in practical use, probably the oldest is the χ^2-statistic. It is just the sum of the squared and normalized deviations between p_d and f_d:

$$\chi^2 = n \sum_{i=1}^{9} \frac{(p_d - f_d)^2}{p_d} \tag{6.11}$$

The χ^2-test is one of the most often used tests in empirical studies about Benford's Law. However, it has a major drawback: if n, the sample size, increases then also χ^2 will grow and as a result the *power* of the test becomes so large that practically always H_0 is rejected.

Alternatives are available, for instance instead of using the normalized squared deviations we may use the so-called *Chebyshev-Distance* which just takes the maximum distance between p_d and f_d:

$$\mu^\star = \sqrt{n} \max_{1 \le d \le 9} |p_d - f_d| \tag{6.12}$$

It should be noted that in performing these types of goodness-of-fit tests we are by no means restricted to the leading digit D_1. The same procedures apply when testing the distribution of the second digit D_2. Its H_0-distribution is given by (6.9). And it is also possible to formulate as null-distribution $P(D_1 = d_1, D_2 = d_2)$, i.e., we test the joint distribution of the first two digits. This is done very ofting in accounting studies. See Section 2, where we will have so say more about this idea.

Tests based on the empirical distribution function

Given a sample $(X_1, X_2, \ldots, X_n)$ of identically and independently distributed random variables with distribution function $F(x) = P(X \leq x)$, the *empirical distribution function (ecdf)* $F_n(t)$ of the sample is defined by

$$F_n(t) = \frac{\text{number of sample values } \leq t}{n} = \sum_{i=1}^{n} \mathbf{1}(X_i \leq 1), \tag{6.13}$$

where $\mathbf{1}(A)$ is the *indicator function* of event A. The ecdf has many remarkable properties, the most important of these being given in the *Glivenko-Cantelli Lemma*:

$$\text{Let } D_n = \sup_t |F_n(t) - F(t)| \quad \text{then } \lim_{n\to\infty} D_n = 0 \quad \text{with probability 1} \tag{6.14}$$

Note that D_n is the maximum absolute deviation between the ecdf and the true distribution function $F(t)$ and the lemma states: this distance becomes arbitrarily small and remains that small as sample size increases. M. Loéve has called (6.14) the *fundamental theorem of statistics*. It implies that the whole unknown probabilistic structure of the sequence X_i can be discovered from data with certainty. Note also the formal similarity of D_n to the *Chebyshev distance* (6.12) introduced above.

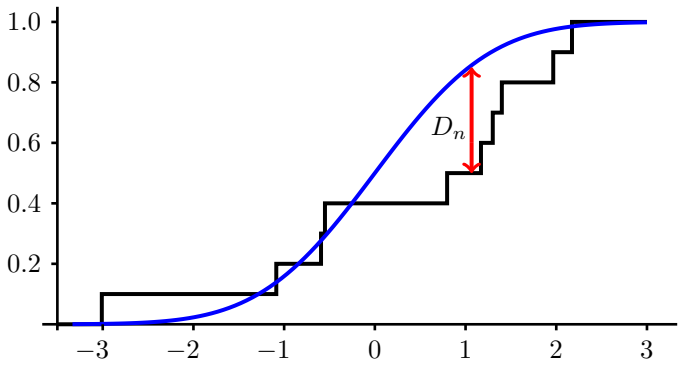


Figure 6.3: The maximum deviation D_n between the ecdf $F_n(t)$ and $F(t)$

Example. In Figure 6.3 I have displayed the ecdf of a sample of size $n = 10$ taken from a standard normal distribution:

$$\mathbf{X} = \{-3.01, -1.09, -0.59, -0.55, 0.79, 1.17, 1.31, 1.42, 1.97, 2.17\}$$

For instance: $F_n(0.5) = 0.4$ because there are 4 sample values ≤ 0.5 and the $F_n(t)$ has jumps of height $1/n = 0.1$ for all sample values are different. The maximum absolute deviation has value $D_n = 0.3852$.

The fundamental result (6.14) gives rise to several classical goodness-of-fit tests. Indeed, D_n is the test statistic of the famous *Kolmogorov-Smirnov Test*: it tests the null hypothesis $H_0 : F(t) = F_0(t)$ against $H_1 : F(t) \neq F_0(t)$ in the two-sided case and rejects H_0 if the observed D_n is too large.

So, it is quite natural to use this test to find out whether observed data deviate significantly from the Benford distribution. However, there is a problem: in the classical setting of the Kolmogorov-Smirnov Test it is assumed that the sample comes from a *continuous distribution*. Of course, the Benford First Digit Law is not continuous, the null distribution is the *step function* given by:

$$F_0(t) = \begin{cases} 0 & \text{for } t < 1 \\ \log(1+d) & \text{for } d \leq t < d+1, \quad d = 1, 2, \ldots, 8 \\ 1 & \text{for } t \geq 9 \end{cases} \tag{6.15}$$

To apply the Kolmogorov-Smirnov Test to a discrete distribution we need a suitably adapted variant of this test to have reliable p-values. Such variants are available, see Arnold and Emerson (2011), and are now part of standard statistics packages like R. I will show you in the next section how to apply these.

An interesting alternative to Kolmogorov-Smirnov ist the *Cramér-von Mises Test*. Here the test statistic is essentially the sum of the *squared deviations* between $F_0(t)$ and the ecdf $F_n(t)$:

$$W^2 = n \int_{-\infty}^{\infty} [F_n(t) - F_0(t)]^2 \mathrm{d}F_0(t) \tag{6.16}$$

You should not worry about the integral occurring here, actually W^2 is a sum because $F_0(t)$ is a step function. Observe the formal similarity between the statistic W^2 and the χ^2-statistics (6.11).

Before applying these statistical tests a decision has to be made which *software* to use. I recommend R, which as I found does a good job regarding our problem. There are two packages which are particularly interesting for or purposes, `BenfordTests` and `dgof` which implement the Kolmogorov-Smirnov and the Cramér-von Mises Tests for discrete distribitions mentioned above and some more. There is also a package `benford.analysis` which may be intersting for your experiments, but it is not discussed here. The functions provided by `BenfordTests` are very handy to use, whereas those of `dgof` are a bit more complicated in use.

6.1.6 Remarkable Properties of Benford's Law

In the sequel I will introduce to you some important and really remarkable results, and I will do so in a rather informal way. The major motivation is

that you should know about these properties of Benford's Law without being incommodated too much by heavy mathematics. Indeed, the derivations and proofs are very technical and require quite sophisticated mathematical tools. The main source is this section is Arno Berger and Theodore P. Hill (2015).

Scale-invariance

A law as universal as Benford's should be *scale-invariant* in the sense that it is independent of units of measurement. In his excellent paper Raimi (1976) writes:

> *. . . that law must surely be independent of the system of units chosen, since God is not known to favor either the metric system or the English system. In other words: a universal first digit law, if it exists, must be scale-invariant.*

Just to give you an example: if accounting data of a big company are Benford in US $, then they should be so too in EUR, British Pounds, etc. This what we expect.

Given a random variable X, scale-invariance means that the distribution of the digits $D_1, D_2, \ldots$ is the same for X and αX for any real scaling constant $\alpha > 0$. Or interpreted in the sense of Benford's law (strong version): the significands $S(X)$ and $S(\alpha X)$ both follow the logarithmic distribution. It is important to note (and source of a common misunderstanding) that scale-invariance refers only to the digit distribution, not to the distribution of X itself. Indeed, no non-null random variable can be scale-invariant.

The scale-invariance property is *characterizing* the law and therefore unique: if for any $\alpha > 0$ and any $d \in \{1, 2, \ldots, 9\}$

$$P(D_1(X) = d) = P(D_1(\alpha x) = d) = \log\left(1 + \frac{1}{d}\right),$$

then X is Benford *et vice versa*, the same is true for (6.10). The Benford distribution is the only distribution with this property.

Powers and products

The power theorem. The classical continuous distributions like normal, uniform, exponential, are not Benford, but something interesting happens if we raise these to powers, Experiment 3 above has given us some indication. Indeed, it can be proved that if X is any continuous random variable having a probability density, then for the sequence X^n, $n = 1, 2, \ldots$ the significands $S(X_n)$ tend in distribution of the logarithmic law (6.6). This result is of considerable value in application, e.g., in fraud detection, see Section 2.

The product theorem. Also, if X and Y are independent random variables and X is Benford, then so is the product XY, provided $P(XY = 0) = 0$. This

result has interesting implications. Consider for instance an inventory stock I_n. Very often these stocks behave like a *random walk process* as goods are added and withdrawn from the inventory in random amounts. We shall see in a moment that random walks are usually *not* Benford, but prices of goods often are. Thus if we want to determine the *value* V_n of an inventory stock at some epoch n we calculate $V_n = I_n \cdot p_n$. So, even when I_n is not Benford but p_n is, then V_n will be Benford.

A limit theorem for products. The nice behavior of Benford's Law with regard to products is also reflected in the following very important result:

If $X_1, X_2, \ldots$ are independent and identically distributed continuous random variables then the significands of their product

$$X_1 \cdot X_2 \cdots X_n = \prod_{i=1}^{n} X_i$$

tend in distribution to Benford's Law.

Here is an example: Let C_0 some initial capital not necessarily random and suppose that we get interest on C_0 with interest rates $r_1, r_2, \ldots, r_n$. Then by compounding interest the value C_n of our capital at epoch n will be:

$$C_n = C_0(1 + r_1)(1 + r_2) \cdots (1 + r_n).$$

If interest rates are continuous random variables then the product limit theorem applies and C_n will be approximately Benford for large n.

Sums

Regarding sums the situation is not so nice as it is for products. Here is an intuitive argument: Benford's Law is a logarithmic distribution. Now $\log(xy) = \log x + \log y$, but it is not possible to express $\log(x + y)$ in simple terms of $\log x$ and $\log y$.

Indeed, if X and Y are both Benford then $X+Y$ will not be Benford. Moreover the following striking result holds:

If $X_1, X_2, \ldots, X_n$ are independent and identically distributed random variables with finite variance, then $\sum_{i=1}^{n} X_i$ is not Bendord in the limit $n \to \infty$, not even a subsequence will be Benford.

As a result, the classical random walk processes $S_n = \sum_{i=1}^{n} X_i$ with increments being discrete with ± 1 or having some other distribution with finite variance are not Benford. An informal argument is this: the conditions stated above are those of the classical Central Limit Theorem. Thus the standardized sum $\sum_{i=1}^{n} X_i$ will tend in distrinution to a standard normal, but the latter can be shown not to be Benford.

However, there is one more *invariance property.*

Sum-invariance. In his PhD-thesis Nigrini (1992) observed that in data sets he considered the *sum of significands* of data points with $D_1 = 1$ was very

close to sums of items with $D_1 = 2$ or any other possible value of the first digit. More precisely, let $\{x_1, x_2, \ldots, x_n\}$ be a data sample of size n and let $S_{d_1,d_2,\ldots,d_m}(x_k)$ be the significand of sample point x_k having the first m digits equal to $d_1, d_2, \ldots, d_m$, otherwise set $S_{d_1,d_2,\ldots,d_m}(x_k) = 0$. By the Law of Large Numbers the arithmetic mean of these significands tends to the mathematical expectation $E[S_{d_1,d_2,\ldots,d_m}(X)]$, i.e.,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} S_{d_1,d_2,\ldots,d_m}(x_k) = E[S_{d_1,d_2,\ldots,d_m}(X)],$$

for all $m \in \mathbb{N}$. So far, nothing special. However, if the data source X is Benford, then this limit is *independent* of the digits $d_1, d_2, \ldots, d_m$, moreover this invariance property is *characterizing* Benford's Law. Indeed, if X is Benford, then it can be shown:

$$E[S_{d_1,d_2,\ldots,d_m}(X)] = \frac{10^{1-m}}{\ln 10} \tag{6.17}$$

for all possible tuples $d_1, d_2, \ldots, d_m$. For $m = 1$ we have $E[S_d(X)] \doteq 0.4343$ for all digit values d, for $m = 2$ $E[S_{d_1,d_2}(X)] \doteq 0.0434$, etc.

Hill's Limit Theorem

Many of the properties discussed so far are *characterizing* Benford's Law, but none of them *explains* its astounding empirical ubiquity. Benford (1938) already observed, many data sets do not conform the law closely, others did reasonably well. But, as Raimi (1976) writes: *what came closest of all, however, was the union of all his tables.* Stated differently: the best fit to the logarithmic law Benford obtained when combining samples coming from so diverse sources like sports results, numbers from newspapers, atomic weights, etc.

This seemingly harmless observation, unnoticed for many years, was the starting point of Hill's seminal work. Theodore P. Hill (1995) derived a new statistical limit law which may be seen as some kind of Central Limit Theorem for significant digits. This limit theorem offers a natural explanation for the empirical evidence of Benford's Law.

Recall, the Central Limit Theorem tells us that under certain mild conditions sums of independent random variables have a normal distribution when the number of summands tends to infinity.

Similarly, Hill's Theorem: *if probability distributions are selected at random and random samples are taken from each of these distributions in any way so that the overall process is scale-unbiased, then the frequencies of significant digits will tend to the logarithmic distribition (6.6) as the sample size* $n \to \infty$.

Some explantions are in order now:

- What does it mean: *a probability distribution is selected at random*?

  Easy (in principle): we perform a random experiment and its result will be a probability distribution. For example, suppose our random experiment

has as possible outcomes two probability distributions F_1 and F_2 forming a sample space $\Omega = \{F_1, F_2\}$. F_1 may be, e.g., a uniform distribution on $[0, 1]$, F_2 a standard normal distribution.

Suppose also that F_1 is selected with probability $1/2$ and this is also the probability for selecting F_2. Think of tossing an unbiased coin[2]: if the coin shows *head* F_1 is selected, otherwise F_2. Once a distribution has been determined a sample of m_1 (say) independent observations is taken from this distribution. The process is repeated, again a distribution is selected at random and becoming this way the source for another sample of m_2 observations. This new sample is *combined* with the first sample to give a larger sample with $m_1 + m_2$ observations. The process continues and stops if our *combined sample* has reached some required size n.

In our example Ω was a set containing only two points, the distribution functions F_1 and F_2. But Ω may be a continuum as well. For instance:

$$\Omega = \{\text{all normal densities with } \mu \text{ and } \sigma^2, \\ \text{where } \mu \text{ has a uniform distribution on } [-\alpha, \beta]\}$$

Here the base experiment is this: first select a random number u uniformly from the interval $[-\alpha, \beta]$. Then take a sample from a normal distribution with $\mu = u$. Repeat this process as long as required.

- What means *the selection process is scale-unbiased*?

  This is not the same as *scale-invariance* discussed above. In fact, it is a much weaker requirement: the sampling process *on average* does not favor one scale over another. It is even possible that none of the distributions in Ω are Benford and therefore scale-invariant. The justification of scale-unbiasedness is somewhat akin to the assumption of independence in context of the Central Limit Theorem. Checking the assumption can be done indirectly by a goodness-of-fit test for the logarithmic law.

6.2 Where to go from here

Having read the *Invitation* you may be now sufficiently motivated to go on reading and see what I want from you.

Writing a *nontrivial* thesis about Benford's Law is certainly a challenge. There are at least two ways to go along.

6.2.1 Statistical Forensics

There is an enourmous amout of literature on the application of Benford's Law to detect data manipulations. Proven cases of such manipulations range from the private sector (financial statements of big companies) over macroeconomic

[2] In technical terms: we are constructing a *random probability measure*.

data reported by governments to EU-authorities to falsification of large sets of clinical data.

Why not write a thrilling case study? You wont have difficulties to find attractive interesting examples of spectacular bankruptcies, of insider trading, you might remember the *Libor Scandal* or diverse illegal manipulations of foreign exchange markets. And, of course, this list is by no means complete.

It has been shown several times that Benford's Law is applicable in diverse auditing contexts including internal, external and governmental auditing. A typical procedure is to test the Benford hypothesis on the first and/or the second digit of data like revenues, canceled checks, inventory and disbursements. Quite often these tests pointed auditors to telltale irregular patterns in various financial transactions. The US Internal Revenue Service uses Benford's Law to sniff out tax cheats and Deutsche Bank crunched the numbers on Russell 3000 companies and found that a Benford distribution applies to almost every balance sheet and income statement item, from annual sales to accounts receivables to net income. The vast majority of companies' data adheres to Benford's Law, with about 5 per cent of Russell 3000 companies not conforming based on Deutsche's calculations. Similar data was found for global firms, see the interesting article in the *South China Morning Post*, July 10, 2015.

Interesting studies have been performed for the public sector. In a series of papers Rauch et al. (2011) studied among others European Union economic data relevant to compliance with Stability and Growth Pact criteria. One of their findings was a significant deviation of Greek official data from Benford's Law. The fact that Greek data manipulation was officially confirmed by EU Commission can be seen as an evidence that Benford's Law might be a valuable forensic tool.

But even sciences and academia are not immune to dishonesty and deception. A well-known case of data falsification from clinical experiments is reported by J. Interlandi in the New York Times Magazine (October 2006), the Poelman Affair. Lee, Tam Cho, and Judge (2015) have shown data manipulated by Eric Poelman show a significant deviation from Benfords's Law.

And finally, another interesting forensic application of Benford's Law is to detect manipulations of elections. Much discussed examples are the election in Turkey, Nov. 1, 2015, and the presidential election in Iran 2009. Remarkably, it is often the *last digit* of vote counts that gives indications to manipulation.

If you decide to pursue the forensic route in your thesis then there are a few points to be taken care of:

- Collect a sufficient amount of data, at least a few hundred items. I know that this is the really cumbersome part of your study. For instance accounting data are often available only via annual reports of companies. These in turn are usually published as pdf-documents. Thus you will have to use appropriate software tools to extract data from those files. Regarding macroeconomic data the situation is much better as the EU offers free public access to most of these data.

- Keep in mind that if your data or parts thereof show significant deviations from Benford's Law this does not automatically mean fraud or data falsification. Indeed, examples are known where unmanipulated data are far from Benford. So, if in such a case it is very likely that your null hypothesis that data are Benford is rejected. Still, it may be possible to use Benford's Law as a forensic tool, *if you transform your data*. Such a transformation is mentioned in Section 7, Remarkable Properties, see also Morrow (2010).
- Formulate your hypotheses carefully and perform various statistical analyses. Regarding tests, please read also the subsection on Experimental Statistics below. In addtion to the methods outlined in Section 5, the R-package `benford.analysis` will be very helpful in this context. Give critical interpretations of your results.

6.2.2 Experimental Statistics

This a second interesting route to follow. What can be said about the *discriminative capabilities* of the statistical tests presented in Section 5, about their *power*? Recall, the power of a hypothesis test is defined as the probability to reject the null hypothesis when the alternative is true. In other words, this is the probability that the test correctly signals data are not Benford. So this is a most important measure of quality of a test.

Determination of the power of a statistical test requires the specification of an alternative. Normally, it is not enough to say: H_A : *data are not Benford*. We must be more specific.

A standard scenario

One possibility is to state as alternative: the distribution of digits is that of uniform distribution (say), a normal distribution, or of an exponential distribution. All these are known to be non-Benford. So, it would be very interesting to estimate the power of the tests described above by means of a systematic *simulation study*.

- What is the effect of the *sample size* on the power?
- How does *variance* influence power? For instance, what is the effect, if we widen the interval of support if a uniform, what if we increase the standard deviation of a normal distribution? I expect that sandard tests of the Benford hypothesis based on digits will show rather different behavior.

Faking data intelligently

Probably the most interesting alternative hypotheses arise, if we try to *fake data in an intelligent way*. What does it mean?

It is no challenge to generate data sets where the first digit follows a Benford

distribution very closely. This requires only a clever application of the properties of the significand function outline in Section 2. For instance you may generate a random sample of uniform variates, take the significands and replace the first digit by digits from a sample following Benford.

What will happen? Data sets manipulated in this way will very likely pass any of the first digit tests, may it be χ^2, Kolmogorov-Smirnov, etc. But how likely is it to reveal the manipulation when testing *the first two digits*? What happens, if we fake the first two digits of the data? How can this be done?

An interesting idea is discussed in Jamain (2001, Section 4): data which are basically Benford are *contaminated* by non-Benford data. How do tests perform in dependence on the amount of contamination?

Now I want you to do something I *never* ask my students: please *activate some criminal energy* in you! Develop ideas, scenarios, models to manipulate data in such a way that the classical digit-based tests perform as badly as much.

Testing characterizing properties

Tests based on the first few significant digits are tests of the weak form of Benford's Law. But can we test also the strong version (6.6)?

This can be done in several ways and it would be very interesting to find out how these approaches perform compared to digit-based tests. Here are some ideas you may consider in your experiments:

- Since the significand function $S(X)$ is a continuous random variable testing (6.6) amount to a classical continuous goodness-of-fit test. Equivalently you may test whether $\log S(X)$ has a uniform distribution on $[0, 1]$.
- What about *scale-invariance*? We know that a random variable is scale-invariant if and only if it is Benford. Smith (1997) has suggested a *Ones-Scaling Test*. It checks the distribution of leading ones in rescaled data, i.e., it examines the relative frequencies of

  $$D_1(X) = 1, \quad D_1(\alpha X) = 1, \quad D_1(\alpha^2 X) = 1, \ldots$$

  where $\alpha > 1$ is some constant such that $\log \alpha$ is *irrational*. For instance $\alpha = 2$ may be a proper choice. How can we test not only leading ones but all possible leading digits simultaneously? Devise such a procedure generalizing Smith's idea.
- What about *sum-invariance*? Only Benford can have this property. When preparing this topic the following idea came into my mind: for a sample $\{x_1, x_2, \ldots, x_n\}$ define

  $$S_d(x_k) = \begin{cases} S(x_k) & \text{if } D_1(x_k) = d \\ 0 & \text{else} \end{cases}$$

  By sum-invariance the *arithmetic means* $\widehat{\mu}_d$ of the $S_d(x_k)$ should have roughly the same values for all possible values of digit $D_1 = \{1, 2, \ldots, 9\}$.

We know from (6.17) its *theoretical* value: if data are Benford, then

$$\mu_d = E(S_1(X)) = 1/\ln 10 \doteq 0.4343.$$

Now by the Central Limit Theorem the standardized values

$$T_d = \frac{\widehat{\mu}_d - 1/\ln 10}{s_d}\sqrt{n}$$

are approximately standard normal for large sample size n, where s_d^2 denotes the sample variance of $S_d(X)$. The sum of squares $\sum_{d=1}^{9} T_d^2$ of the statistics T_d follows a χ^2 distribution with 9 degrees of freedom. So we can use this distribution to determine critical values and p-values and thus have one more test of the Benford hypothesis. However, my argument has a weakness: it requires *independence* of the T_d. Suppose, we tacitly ignore this point, how does this sum-invariance test perform?

6.3 An Annotated Bibliography

Searching the web with google using *Benford's Law* as a search key I get at the time of writing this more than 100 000 hits. There is an enormous amount of publications dedicated to this law and it looks to me as if this number is growing exponentially.

The papers and books on Benford's Law fall roughly into two categories: (i) theoretical (extensions of the law, conditions when it does hold, when it does not, special topics from probability, number theory, computer science), (ii) applications (forensic statistics, auditing, social sciences, astronomy). There is an extensive online bibliography created and maintained by A. Berger, T. P. Hill, and E. Rogers (2016) covering all these categories.

Strongly recommended is the fine book edited by S. Miller (2015). This textbook has six parts, the first two of them devoted to the mathematical theory of Benford's Law, the other parts cover many interesting applications from fraud detection, diagnosis of elections, applications in economics, psychology, natural sciences and computer science.

Regarding the mathematical theory the most comprehensive text and standard reference is the book by Arno Berger and Theodore P. Hill (2015). Although sometimes technically demanding it gives a very readable and excellent coverage of the current state of the art. Also recommended regarding theory is the master thesis by Jamain (2001).

If you plan to work on statistical forensics then there is no way around reading Nigrini. After his PhD Thesis (Nigrini, 1992) he has published quite a number of papers on applications of Benford's Law to fraud detection, auditing and accounting. He has authored also a very interesting book (Nigrini, 2012) on the subject. Here you find several examples and demonstrations of statistical tests partly developed by the author and implemented in a spread sheet calculator.

6.4 References

[1] Taylor A. Arnold and John W. Emerson. "Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions". In: *The R Journal* 3.2 (2011), pp. 34–39. URL: http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Arnold+Emerson.pdf.

[2] Frank Benford. "The law of anomalous numbers". In: *Proc. Amer. Philosophical Soc.* 78 (1938), pp. 551–572.

[3] A. Berger, T. P. Hill, and E. E. Rogers. *Benford Online Bibliography*. 2016. URL: http://www.benfordonline.net.

[4] Arno Berger and Theodore P. Hill. *An Introduction to Benford's Law*. Princeton University Press, 2015.

[5] William Feller. *An Introduction to Probability Theory and Its Applications*. 2nd. Vol. 2. John Wiley and Sons, 1971.

[6] Theodore P. Hill. "A statistical derivation of the significant-digit law". In: *Statist. Sci.* 10.4 (1995), pp. 354–363.

[7] Theodore P. Hill. "The difficulty of faking data". In: *Chance* 12.3 (1999), pp. 27–31. URL: http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1048&context=rgp_rsr.

[8] Adrien Jamain. "Benford's law". MA thesis. Imperial College of London, 2001.

[9] J. Lee, W. K. Tam Cho, and G. Judge. In: Stephen J. Miller. *Benford's Law: Theory and Applications*. Ed. by Stephen J. Miller. Princeton University Press, 2015. Chap. Generalizing Benford's Law.

[10] Stephen J. Miller. *Benford's Law: Theory and Applications*. Ed. by Stephen J. Miller. Princeton University Press, 2015.

[11] John Morrow. *Benford's Law, Families of Distributions and a Test Basis*. http://www.johnmorrow.info/projects/benford/benfordMain.pdf. last accessed Feb 25, 2016. 2010.

[12] Simon Newcomb. "Note on the Frequency of Use of the Different Digits in Natural Numbers". In: *Amer. J. Math.* 4.1-4 (1881), pp. 39–40. URL: http://dx.doi.org/10.2307/2369148.

[13] M. J. Nigrini. *Benford's Law: Applications for Forensic Accounting, Auditing, and Fraud Detection*. Wiley Corporate F&A. John Wiley & Sons, 2012.

[14] M. J. Nigrini. "The detection of income tax evasion through an analysis of digital distributions". Thesis (Ph.D.) Cincinnati, OH, USA: Department of Accounting, University of Cincinnati, 1992.

[15] Ralph A. Raimi. "The first digit problem". In: *Amer. Math. Monthly* 83.7 (1976), pp. 521–538.

[16] Bernhard Rauch et al. "Fact and Fiction in EU-Governmental Economic Data". In: *German Economic Review* 12.3 (2011), pp. 243–255. URL: http://dx.doi.org/10.1111/j.1468-0475.2011.00542.x.

[17] S. W. Smith. "The Scientist and Engineer's Guide to Digital Signal Processing". In: Republished in softcover by Newnes, 2002. San Diego, CA: California Technical Publishing, 1997. Chap. 34 - Explaining Benford's Law.

Topic 7

The Invention of the Logarithm

A Success Story

For it would be without doubt an incredible stain in analysis, if the doctrine of logarithms were so replete with contradictions that it were impossible to find a reconciliation. So for a long time these difficulties tormented me, and I was under several illusions concerning this matter, in order to satisfy myself in some manner without being obliged to completely overturn the theory of logarithms.

Leonhard Euler, 1749

Keywords: *history of mathematics, logarithmic and exponential function numerical mathematics*

7.1 An Invitation

7.1.1 A personal remembrance

In the early 1970s I was attending secondary school in Vienna (Astgasse). The mathematics courses there were pretty demanding and I clearly remember that one of the most cumbersome affairs in these courses was to perform various numerical calculations. We had to do these *by hand*, sometimes with the aid of a *slide rule*. The use of the latter required a good deal of dexterity and skill and getting correct results was not only a matter of meticulous preciseness but also of good luck[1].

This annoying situation changed in the sixth class. At the beginning of that year every pupil was handed a small innocuous booklet, *Vierstellige Logarithmen*. After having browsed through the book, my opinion was: I have never seen such an boring book! There were tables after tables, practically no text, only tables on each page. However, a few weeks later I had to modify my opinion. After having been introduced to the concept of the exponential function and its inverse, the logarithmic function, after having been taught how to solve simple

[1] At that time I didn't know that slide rules use essentially logarithmic scales!

exponential equations, our instructor gave us a brief course on how to use the tables of logarithms. It turned out that only a few rules had to be obeyed:

$$\log(a \cdot b) = \log(a) + \log(b), \quad \log\frac{a}{b} = \log(a) - \log(b),$$
$$\log(a^b) = b\log(a), \quad \log(1) = 0 \tag{7.1}$$

It was a quantum leap! Multiplying two numbers? Easy, just determine their logarithms by a table look up, add these and one more table look up gives the result[2]. Division of numbers? What a nerve racking and tedious task when done by hand! But with logarithms it is as easy as subtracting two numbers! Calculating powers and roots? Also easy, it's just a division! Needless to say, that I was really enthusiastic about this new tool, and my enthusiasm was shared by most of my class mates[3].

In my last year at secondary school the first pocket calculators became available, Texas Instruments was the leading company, offering the scientific calculator TI SR 50. It was as large as a brick stone and almost as heavy as a brick stone. And it was extraordinary expensive! The price of such a handy computer was higher than the average monthly salary of a worker at that time. But the capabilities of these small computers were really amazing. Now you could do all that hard numerical stuff without resort to logarithm tables. Not surprisingly, at the beginning of the 1980s tables of logarithms were completely replaced by pocket calculators, since their prices have gone down drastically. An era ended at that time, a truly remarkable success story which has lasted more than 350 years.

But how did it begin?

7.1.2 Tycho Brahe - the man with the silver nose

Tycho Brahe, born on 14 December 1546, originated from a famous Danish noble family. At an age of only 13 years he started studying at the University Copenhagen. The solar eclipse in 1560 which has been predicted with high accuracy inspired him to concentrate his studies on astronomy. He continued his studies in Leipzig and Rostock. There, in 1566 he got involved into a heavy dispute with another Danish noble man (rumors say that the dispute was about a mathematical formula). The dispute ended in a duel with sabers in which Tycho's nose was cut off. That accident disfigured him for the rest of his life. In part this disfacement

Tycho Brahe
1546–1601

[2] Actually, some intermediate scaling steps are necessary, but as our logarithms were to base 10, these were also quite easy.

[3] A few years later at the university I was taught the basic principles of *Laplace Transforms*, another key experience, which reminded me of logarithms. Using Laplace Transforms, differentiation of a function becomes (essentially) a multiplication by a variable, a definite integral is a simple division, etc.

could be hidden by an artificial nose made of silver. For this reason Tycho became known as *the man with the silver nose*.

Tycho's studies and later his scientific work as an astronomer were characterized by his efforts to collect astronomical data from very accurate measurements and these in turn required high precision instruments.

In 1575, at an age of 21 years, he was already a renown and eminent scientist and planned to leave Denmark for Basel. King Frederick did not want to lose him and offered him the island Ven located in the Öresund between Sweden and Denmark where he built Uraniborg Castle, a combination of residence, alchemistic and technical laboratories and astronomical observatories. There he lived together with his court jester, a dwarf named Jepp, and his moose[4] and pursued deep and comprehensive scientific studies. This work was interrupted from time to time by fabulous festivities. Indeed, at Uraniborg opulent banquets were held regularly for illustrious guests and Tycho proved to be a charming and entertaining host.

It happened in the fall of 1590 that King Jacob VI of Scotland (later King James I of England) sailed with a delegation to Denmark to meet his bride-to-be, Anna of Denmark. Due to very bad weather the royal society was forced to land on Ven, not far away from Uraniborg, where they found shelter for a few days. Tycho showed himself from his best side as entertainer and host and on that occasion he told Dr. John Craig (?–1620), personal physician of King James, about a recent, truly marvelous invention, *prostaphaersis*. With its help extremely complex and expensive astronomical calculations could now be carried out with breathtaking ease.

7.1.3 Prostaphaeresis

This tongue-twisting word is a composition of the two Greek words *aphairein* and *prostithenai* which mean to subtract and to add, respectively. The basis of this remarkable method is formed by classical addition theorems for trigonometric functions. For instance, it has been known since ancient times that

$$\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \tag{7.2}$$

$$\cos(\alpha-\beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta). \tag{7.3}$$

These formulas can be proved by elementary geometry or, more illuminating, by using *Euler's Equation*, see Section 2 below. If we add (7.2) and (7.3) we obtain after simplification:

$$\cos(\alpha)\cos(\beta) = \frac{\cos(\alpha+\beta)+\cos(\alpha-\beta)}{2}. \tag{7.4}$$

Looking closer at (7.4) you may realize a remarkable fact: the left hand side is a multiplication of two numbers, namely the product of two cosines, whereas

[4]The moose, kept as a house pet, was allowed to move freely inside Uraniborg and had the somewhat strange and not species-appropriate attitude to drink lots of Danish beer. One day the moose being heavily drunk dropped down the staircase and broke its neck.

on the right hand side we have essentially a summation (scaled down by two). Johannes Werner (1468-1522), a German mathematician and astronomer was one of the first to realize the potential of this formula to reduce the burdening and cumbersome multiplication of numbers to much easier addition. All one needed to exploit this potential was just a comprehensive collection of tables of the sine and cosine functions. But such tabular material was already available at that time, rather voluminous collections of tables giving the sine, cosine, tangent and secant ($1/\cos(\alpha)$) functions with an accuracy of 10 decimal places and even more.

To see how the method works let me give you an example. However, we will not use tables but rather a pocket calculator. The point is to *see* how prostaphaeretic multiplication is done.

Suppose, we want to calculate the product $17 \cdot 258$. Since sine and cosine functions take their values in the interval $[-1, 1]$ we scale first:

$$17 \cdot 258 = 100 \cdot 0.17 \cdot 1000 \cdot 0.258 = 10^5 \cdot 0.17 \cdot 0.258$$

Then we find angles α and β such that

$$\cos(\alpha) = 0.17, \quad \cos(\beta) = 0.258$$

These angles were determined formerly by look up in the tables, today we find them using the arccos-function of a pocket calculator:

$$\alpha \simeq 80^\circ\ 12'\ 43'', \qquad \beta = 75^\circ\ 2'\ 56''$$

Of course, using radians instead of degrees would be a bit more comfortable, but let us follow the way people worked at that time as closely as possible. Next we apply Werner's formula (7.4):

$\alpha+\beta$	$=$	$155^\circ\ 15'\ 39''$	$\cos(\alpha+\beta)$	$=$	-0.90928
$\alpha-\beta$	$=$	$5^\circ\ 9'\ 47''$	$\cos(\alpha-\beta)$	$=$	0.99594
					0.08766
$\times 0.5$				$=$	0.04386
$\times 10^5$				$=$	4386

which is indeed the correct result $17 \cdot 258 = 4386$.

Observe, that only addition and scaling (simply a shift of the decimal point) are necessary.

Division is also easy. Suppose we want to calculate x/y. Rewrite this as $x \cdot 1/y$ and put $x = \cos(\alpha)$ and $1/y = \cos(\beta)$. From the latter we have $y = \sec(\beta)$, the *secant*-function, which was also extensively tabulated. Now use again Werner's formula (7.4).

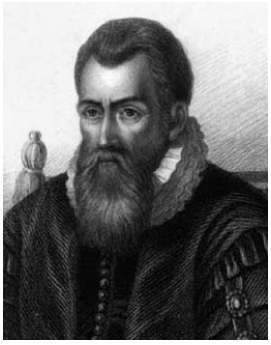
Although nowadays prostaphaeresis appears to us as a rather laborious method, at Tycho's and Kepler's time it was considered a major breakthrough in computational mathematics because it reduces the hard work of multiplication and division to the much simpler operations of addition and subtraction. Therefore it is not surprising that it found broad dissemination in Europe because it

simplified computational work so much, especially in astronomy and nautical navigation.

John Craig was deeply impressed by Tycho's explanation of the new method. Incidentally he was not only physician but has also studied astronomy in Germany and one of his teachers was Paul Wittich (1546-1586), in later years assistant of Tycho. When Craig returned to Scotland he contacted his friend *John Napier* and told him about this wonderful invention of prostaphaeresis, the *Artificium Tychonicum*, as Johannes Kepler once called it.

7.1.4 John Napier and Henry Briggs

That John Craig was friend of John Napier is one of those remarkable incidences in the history of science which often gave the impetus to new and important developments. John Napier, Earl of Murchiston, was an affluent Scottish laird, mainly busy administering his estates. Science was his leisure time activity. He concentrated on various disciplines like theology and mathematics. Thus he was not a professional mathematician but a very talented *amateur*. Computational mathematics was a topic he was most interested in. For instance, he invented the *Napier Rods*, wooden tablets which could be used to multiply numbers and even calculate square roots. Indeed, in 1614 Napier wrote: *I have always endeavoured according to my strength and the measure of my ability to do away with the difficulty and tediousness of calculations, the irksomeness of which is wont to deter very many from the study of mathematics.*



John Napier
1550-1617

At the time when Craig informed him about the new method of prostaphaeresis he was for some years already thinking about ways to reduce multiplication do addition, division to subtraction. Craigs message prompted him to increase his efforts.

Interestingly, he did not develop further prostapaeresis, as we might expect, instead he based his approach on an idea which goes actually back to Archimedes of Syracuse (287? - 212? BC). It is the idea of *correspondence* between arithmetic and geometric progressions.

Let me give an example of such a correspondence: let $n = 0, 1, 2, \ldots$ denote an arithmetic progression, it's simply the sequence of non-negative integers. To each term in this sequence define $a_n = 2^n$, a geometric progression with initial value 1 and common ratio 2. For $n = 0, 1, 2, \ldots, 10$ the correspondence is conveniently given in the following table:

n	0	1	2	3	4	5	6	7	8	9	10
a_n	1	2	4	8	16	32	64	128	256	512	1024

Now let us call n, the numbers in the first row, the *logarithms* of a_n, the numbers

in the second row. We can easily check by example, that this overly simplistic table is indeed a table of logarithms satisfying the basic log-rules (7.1). For instance to multiply 8 by 128 we would use our table this way:

$$\log(8 \cdot 128) = \log(8) + \log(128) = 3 + 7 = 10,$$

but $10 = \log(1024)$, thus $8 \cdot 128 = 1024$. Also division is easy with our table. Suppose we want to calculate $512/64$, then by applying the division rule (7.1) and table look up:

$$\log \frac{512}{64} = \log(512) - \log(64) = 9 - 6 = 3,$$

but $3 = \log(8)$, therefore $512/64 = 8$. What about a square root, $\sqrt{256}$, say? Easy again,

$$\log(\sqrt{256}) = \log(256^{1/2}) = \frac{1}{2}\log(256) = \frac{1}{2} \cdot 8 = 4.$$

From our table we obtain $4 = \log(16)$ which implies $16 = \sqrt{256}$. Thus all log-rules apply. That's fine.

But now a problem appears. Suppose, we want to calculate $\sqrt{128}$ by means of our table. Proceeding as before, we find $\log(\sqrt{128}) = 3.5$, but our table has no entry a_n for $n = 3.5$. Also, the table doesn't have entries n, i.e. logs, for the integers between 8 and 16, neither for those between 16 and 32, etc. There are *gaps* in the table! And these gaps become progressively larger. Thus in this simple layout, our log-table turns out to be not very useful, it is not sufficiently *dense*.

Napier was certainly aware of this problem and constructed his tables using a geometric progression in which the terms a_n are very close together, thus are much denser. His solution was so simple that the world wondered why no one had thought of it before, as Pierce (1977) remarks.

In modern notation the arithmetic and geometric progressions Napier used were:

n	0	1	2	...	m	...
a_n	v	$v(1-1/v)$	$v(1-1/v)^2$	...	$v(1-1/v)^m$	...

The number v was chosen by Napier to be $v = 10^7$. This choice was mainly inspired by the major applications Napier had in mind, numerical calculations with values of trigonometric functions. That's also the reason why Napier called the a_n *sines*. His choice of the ratio $1 - 1/v$ was a rather clever one as it results in a *very slowly decreasing* sequence. Indeed, we find:

$$\begin{aligned}
0.9999999^0 \cdot 10^7 &= 10^7 \\
0.9999999^1 \cdot 10^7 &= 9\,999\,999 \\
0.9999999^2 \cdot 10^7 &= 9\,999\,998.0000001 \\
0.9999999^3 \cdot 10^7 &= 9\,999\,997.0000003 \\
&\ldots \\
0.9999999^\alpha \cdot 10^7 &= A \\
&\ldots
\end{aligned}$$

This is essentially Napier's First Table. We call the exponent α in the last line above the *Napier Logarithm* of A and denote it by $LN(A) = \alpha$. Thus we define (as Napier did) the logarithm of A by

$$(1 - 1/v)^{\alpha} v = A \Leftrightarrow \alpha = LN(A), \quad \text{where } v = 10^7. \tag{7.5}$$

Napier himself introduced the term *logarithm* as a synthesis of the Greek words *logos* and *arithmos* meaning ratio and number, respectively.

The calculation of the numbers in this table is greatly facilitated by the fact that computation can be performed *recursively* and only subtractions are necessary. To see this, observe that

$$\begin{aligned} a_n = (1 - 1/v)^n v = (1 - 1/v)^{n-1} v(1 - 1/v) = \\ = a_{n-1}(1 - 1/v) = a_{n-1} - 10^{-7} a_{n-1}, \end{aligned}$$

but $10^{-7} a_{n-1}$ is merely a shift of the decimal point. Still, the amount of computational work mastered by Napier was really impressing. Later Napier constructed two more tables based on the first table to cover a broader range of values.

It is important to observe that the Napier logs do not satisfy the standard rules (7.1). In particular, there is no *basis* β in this system (as e.g., the Eulerian number e is basis of the natural logarithms). Moreover, $LN(1) \neq 0$. Indeed, it is a huge number which can be shown to be (see Section 2):

$$LN(1) \doteq 161\,180\,948.53537\,38070 \tag{7.6}$$

Here and in the sequel the symbol `doteq` $\doteq$ means that the right hand number is given correctly in all displayed decimal places. In (7.6) therefore $LN(1)$ is correct to 10 decimal places.

As a result the all important multiplication rule actually is:

$$LN(A \cdot B) = LN(A) + LN(B) - LN(1), \tag{7.7}$$

and similar adaptations are necessary to the division- and power rule. So Napier's system of logarithms is a system which one has to get used to. But once one has acquired some fluency with the rules of Napier's system practical calculations using his tables could be performed rather routinely.

After almost twenty years of incredible hard work 1614 John Napier published his tables in a book entitled *Mirifici Logarithmorum Canonis Descriptio*. In 1619 a second book, *Mirifici Logarithmorum Canonis Constructio*, was published posthumously which gives a description of the method he had used to calculate his tables.

Napier's publications were almost immediately accepted and appreciated by the scientific community of that time. It is only legitimate to say that Napier's logarithms represented a major break-through in computational mathematics.

Henry Briggs (1561–1630) came across Napier's 1614 *Canon* almost immediately after its publication. At that time he was professor of geometry at

Gresham College, London. He began to read it with interest, but by the time he has finished, his interest was changed into enthusiasm. The book was his constant companion: he carried it with him when he went abroad; he conversed about it with his friends; and he expounded it to his students who attended his lectures, as Thompson and Pearson (1925) reported. Briggs decided to leave London for Scotland to visit Napier. When he arrived at Napier's house Briggs addressed him full of deep admiration (Cajori, 1913a):

My Lord, I have undertaken this long journey purposely to see your person, and to know by what engine of wit or ingenuity you came first to think of this most excellent help in astronomy viz., the logarithms.

Briggs didn't come empty-handed. Indeed, he had a lot of ideas and suggestions to improve the wonderful invention. He remained there as Napier's guest for about a month and during that time in many fruitful discussions and conversations the concept of the *common logarithm* was born, i.e. the logarithms with base 10. This idea improved on Napier's own first construction substantially, because now the system had a base, so that :

$$\log_{10}(1) = 0 \quad \text{and} \quad \log_{10}(10) = 1.$$

As a result, the somewhat clumsy rules for Naperian logs where simplified considerably. For instance the essential multiplication property becomes

$$\log_{10}(\alpha \cdot \beta) = \log_{10}(\alpha) + \log_{10}(\beta),$$

in contrast to (7.7), because now $\log_{10}(1) = 0$. In this new system of common logarithms all our standard rules (7.1) hold.

Back to London Briggs immediately started calculating the new logarithms. He presented his results to Napier in 1616 on occasion of a second visit to Scotland. Soon after that meeting Napier died. In 1619 Briggs moved ahead in his career and became first Savillian Professor of Geometry at the University of Oxford. During the following five years he carried on his computational work, and in 1624 he published his famous book *Arithmetica Logarithmica* which contained the common logarithms of 30 000 numbers, the values given with an accuracy of 14 (!) decimal places. This book also has an excellent introduction into new methods and techniques Briggs had to develop to perform his incredibly messy computations. Soon afterwards, in 1628, based on Briggs' work the Dutchman *Adriaan Vlaq (1600–1667)*[5] published tables of common logarithms of the numbers 1 - 100 000 with an accuracy of 10 decimal places. By 1630, when Briggs died, logarithms have been widely accepted and disseminated all over Europe as a most marvelous tool for numerical computations in so diverse fields like physics, engineering, astronomy and especially nautical navigation. The great french astronomer and mathematician *Pierre-Simon Laplace (1749–1827)* brought it to the point when asserting that Napier and Briggs *by shortening the labours (of calculation) doubled the life of the astronomer.*

So it is not an exaggeration to say: The work of Napier, Briggs and their successors on logarithms has given rise to an almost unparalleled success story!

[5] Vlaq had two professions, he was a surveyor and also a quite successful book publisher.

7.2 Where to go from here

This very short introduction into the history of logarithms is meant to be a starting point for a deeper study of this interesting subject.

If you have read other topics in this book you may remember that usually at this place I present various suggestions divided into *mandatory* and *optional*, i.e., suggestions which you should or may take care of in your thesis.

As this topic is a rather special one I have divided my suggestions into those which emphasize the historical perspective and those which are of a more technical flavor.

In designing your thesis you may:

- Put your emphasis on history, or
- You may concentrate on some technical issues, e.g., how to calculate logarithms numerically. Or, how to define logarithms for negative and complex numbers.
- Or, as a third possibility, you may try to find a way to bring together both aspects in a fine and interesting way.

So, make up your mind and read on. And, of course, bear in mind, *your own ideas are always welcome*!

7.2.1 Historical Issues

1. Decimal fractions and mathematical notation

The time around the end of the 16th century and the beginning of the 17th century was a transitional period in the history of mathematics. The foundations of many wonderful discoveries notably the invention of the differential and integral calculus were laid at that time. Two important innovations are directly connected to the works of Napier and Briggs: (a) the development of the modern exponential notation and (b) the propagation of *decimal fractions*. Indeed, Napier seems to be the first to use the *decimal point* systematically. You will find interesting material about these issues at various places in Boyer and Merzbach (2011), also Cajori (1913b) is a valuable source.

2. Properties of Naperian Logarithms

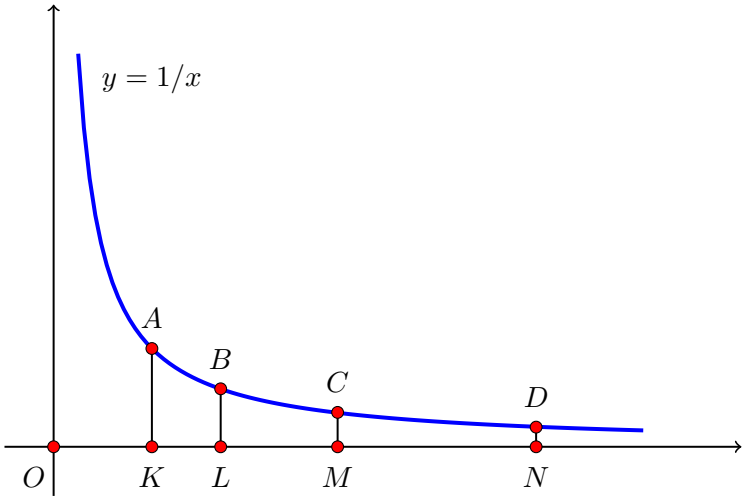
Give a careful exposition of the mathematical properties of Napier's logarithms. I recommend that you start with the definition (7.5) and write $LN(A)$ in terms of *natural logs*. In this way you can readily determine the numerical value (7.6) of $LN(1)$ and formulate analogues to our common rules (7.1). You will find that by a proper scaling the rules for multiplication, division and powers are not so different from (7.1). Furthermore, it would be fine if you could elaborate on a *geometric device* used by Napier to define his logs. The papers of Ayoub (1993) and Panagiotou (2011) will be very helpful in this context.

3. Other people working on logarithms at Napier's time

Raymond Ayoub (1993) writes that the invention of the logarithms in 1614 *is one of those rare parthogenic events in the history of science–there seemed to be no visible developments which foreshadowed its creation.* But still, it is true that several contemporaries of Napier and Briggs have been working on very similar concepts, just to mention *Jobst Bürgi* and *John Speidell.* Others, like Johannes Kepler worked at their own system of logarithms, being inspired by the works of Napier and Briggs. Give a brief account of the approaches these people pursued.

4. The natural logarithm

Recall the original idea of John Napier: construction of logarithms based on a correspondence between a geometric and an arithmetic progression, where the geometric progression should be sufficiently *dense.* In 1647 the Belgian Jesuit Gregory of St. Vincent published a study about an interesting property of the *algebraic curve* $xy = 1$, which is a hyperbola, see the figure given below.



He observed and proved by geometric arguments:

If the line segments OK, OL, OM, ON form a geometric progression, thus if $|OK| = \alpha > 1$:

$$|OL| = \alpha^2, \quad |OM| = \alpha^3, \quad |ON| = \alpha^4, \ldots$$

Then the areas

$$(ABLK), \quad (BCML), \quad (CDNM)$$

are all equal. But this in turn means that the areas

$$(ABLK), \quad (ACMK), \quad (ADNK)$$

form an *arithmetic progression*! Thus we have again a *correspondence* between an arithmetic and a geometric progression. However, such a correspondence is the basic principle of any logarithmic system. But now, in this special case, logarithms have a very *natural* meaning, as they represent areas of certain geometric figures.

Still, the question remains: is this observation helpful at all? It converts a difficult concept into another difficult concept, namely solving a *quadrature problem*, i.e., finding the area below a hyperbolic curve. Today we know that such quadratures can be solved by means of integral calculus.

In 1668 *Nicolaus Mercator (1620–1687)* developed an entirely new approach to the aforementioned quadrature problem thereby finding a way to determine the values of *natural logarithms*, as he called them.

He considered the algebraic curve $(x+1)y = 1$, equivalent to $y = \frac{1}{x+1}$. By long-division he obtained the non-terminating series

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots \tag{7.8}$$

At Mercator's time it was already known that for integers $n \neq -1$

$$\int x^n \,\mathrm{d}x = \frac{x^{n+1}}{n+1} + C$$

Then he integrated (7.8) term by term to obtain:

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \tag{7.9}$$

Using this series he evaluated $\ln(1.1)$ approximately by setting $x = 0.1$ in (7.9). Truncating the series after a few terms he arrived at

$$\ln(1.1) \simeq 0.1 - \frac{0.1^2}{2} + \frac{0.1^3}{3} - \frac{0.1^4}{4} + \frac{0.1^5}{5} = 0.095310,$$

which is correct to 6 decimal places.

Of course, this interesting story does not end her, nor is it complete. You are invited to fill the gaps in this short exposition and find out more. For instance:

- How could Mercator know that the antiderivative of x^n equals $x^{n+1}/(n+1)$?
- Is it always possible (in the sense of *allowed*) to integrate an *infinite series* term by term?
- Will the Mercator series work for *all* values of x?

And truly interesting from a historical point of view: This idea of series integration played a very important role when differential and integral calculus have been invented, in particular it figured prominently in the Leibniz-Newton calculus controversy, a most famous dispute over priority in the development of calculus. I suggest that you consult the papers of Panagiotou (2011) and Burn (2001), but see also Chapter 5 of Sonar (2016).

7.2.2 Technical Issues

5. How did Briggs calculate his tables?

This is a really interesting question! You should bare in mind that at the time of Napier and Briggs several important mathematical concepts have not been known. The idea of a *function* was not yet available, not to talk about exponential and logarithmic functions and their relation to each other. And it took another fifty years until differential and integral calculus have been invented.

So, Briggs had to perform his extensive calculations without the help of powerful mathematical tools. This lack of mathematical machinery he filled with extraordinary diligence and ingenuity. Regarding ingenuity, Briggs not only anticipated *Newton's binomial theorem* (discovered around 1664), he invented also the *calculus of finite differences*, a class of powerful and fascinating methods to deal with *sequences* of numbers. You will find the work of Denis Roegel (2010) to be very helpful.

6. How are logarithms calculated today?

The development of high speed computers during the last decades has rendered tables of logarithms essentially obsolete. So why to bother about methods to calculate logarithms numerically?

Well, simply because logarithms are ubiquitous. There is a vast number of formulas in practically all areas of mathematics and its applications which contain logarithms. Or, think about the rather elementary task of solving an exponential equation. Also, when dealing with very large numbers, so large that computers run into trouble, logarithms come to our help. A striking example is the *factorial function* $n! = 1 \cdot 2 \cdot 3 \cdots n$ which grows extremely fast. But taking logs numbers can be kept at a manageable size, since the logarithmic function is growing rather slowly[6]. For instance, if $n = 100$, then $\ln(100!) \doteq 363.74$, but $100! \approx 10^{158}$, this makes a big difference.

But how do computers calculate logarithms?

One approach is to use *power series expansions*. Several useful series are known, the oldest being the Mercator series (7.9). The latter, however, is convergent only when $-1 < x \leq 1$, and for values x which are close to 1 in absolute value, (7.9) and other series of this type converge very slowly and become practically useless in this case.

Find other power series which are suited for the calculation of logarithms! You will find out that there are a lot of them. The *Taylor-McLaurin Theorem* will be very helpful. So, you should be able to use this tool to find other series.

[6]Sometimes also iterated logs are used, e.g. in number theory or probability. These are functions of the form $\ln\ln(x)$ or even $\ln\ln\ln(x)$. The famous number theorist Carl Pomerance (1944–) once humorously remarked: $\ln\ln\ln(x)$ *goes off to infinity with* x *but has never been observed to do so.*

Discuss also *convergence issues* of the series you suggest. In this context, it is good to know about the following *trick*: any real number $x > 0$ can be written as $x = 2^m y$, where m is an integer and

$$\frac{1}{\sqrt{2}} < y \leq \sqrt{2}.$$

So, $\ln(x) = m\ln(2) + \ln(y)$. Setting $y = 1 + z$, using the above bounding you always have $|z| < 1$.

An alternative to series are *rational approximations*, i.e. fractions of polynomial expressions. A striking example is this one which is also known as *Shank's approximation*:

$$\ln\left(\frac{1+x}{1-x}\right) \simeq \frac{2x(15-4x^2)}{15-9x^2} \tag{7.10}$$

To approximate $\ln(a)$ just put

$$\frac{1+x}{1-x} = a \quad \Longrightarrow \; x = \frac{a-1}{a+1}$$

For instance, to approximate $\ln(2) = 0.69314718\ldots$, set $x = 1/3$ and obtain $\ln(2) \simeq 0.6931216931217$ which is correct to four decimal places.

Approximations like (7.10) often (though not always) have their origin in a *continued fraction*. One well-known continued fraction for the natural logarithm is

$$\ln\left(\frac{1+x}{1-x}\right) = \cfrac{2x}{1-\cfrac{x^2}{3-\cfrac{4x^2}{5-\cfrac{9x^2}{7-\cfrac{16x^2}{9-\cfrac{25x^2}{11-\ldots}}}}}} \tag{7.11}$$

To make use of expressions like (7.11) we terminate the continued fraction early, after the first, the second, etc. partial denominator, by dropping subsequent terms. In this way we obtain a series of *approximants* which become successively more accurate:

$$\ln\left(\frac{1+x}{1-x}\right) \simeq \frac{2x}{1} = 2x \qquad \text{1. approximant}$$

$$\simeq \frac{2x}{1-\dfrac{x^2}{3}} = \frac{6x}{3-x^2} \qquad \text{2. approximant}$$

$$\simeq \cfrac{2x}{1-\cfrac{x^2}{3-\cfrac{4x^2}{5}}} = \frac{2x(15-4x^2)}{15-9x^2} \qquad \text{3. approximant}$$

etc.

As you can see, the 3rd approximant already equals Shank's approximation. You are invited to give these approximations a try, just take your pocket calculator and check various values, e.g., $\ln(2), \ln(10)$, etc.

This was just one spectacular example, several other continued fractions for logs are known.

The theory of continued fractions is a really fascinating area of mathematical analysis, but frankly speaking, it is also rather difficult. If you want to learn more about them you may consult Jones and Thron (1980). In chapter 6 of this book you can find (7.11) as a special case of a more general result.

There are many other powerful approaches to calculate logs. An extremely efficient algorithm is based on the concept of *arithmetic-geometric mean*, you may have a look at the paper of Carlson (1972).

7. Logarithms of negative and imaginary numbers

This suggestion is interesting both historically and from a technical point of view. You should have some basic knowledge of complex numbers*, as it is presented in typical textbooks for undergraduates.*

By the end of the 17th century logarithms, common (base 10) or natural, were well established as an invaluable tool for computational mathematics. In an expression like $\ln(x)$ or $\log_{10}(x)$, x was always considered a positive number. It was also known that the logarithmic function and the exponential function are inverses to each other. Since $y = 10^x > 0$ for all x and because $x = \log_{10} y$, nobody felt the need to consider logarithms of negative numbers.

Johann Bernoulli
1667–1748

Still, at the beginning of the 18th century a remarkable debate started about the question, what sense should be given to the expression $\log(-a)$ for a positive real number a. Here log denotes the logarithm to *any* basis. In a remarkable correspondence in the years 1712-1713 Gottfried W. Leibniz and Johann Bernoulli discussed this problem. Unfortunately these letters were not published before 1745.

But why this discussion? Cajori (1913b) gives an explanation. In the 18th century there was the tendency to take rules derived only for a special case of a mathematical concept and apply them to more general cases. This tendency became more and more pronounced and was called the *principle of the permanence of formal laws*. So by this principle or simply guided by *scientific curiosity* the question of logs of negative numbers became more and more interesting.

The controversy between Leibniz and Bernoulli did not result in a satisfactory answer. On the contrary, quite disturbing contradictions were found when extending the concept of a logarithm to negative numbers. Indeed, negative

numbers themselves were not generally accepted at that time. For instance, the renown french mathematician *Blaise Pascal (1623–1662)* regarded the subtraction of 4 from 0 as pure nonsense (Kline, 1980, pp 114-116)! It seemed to many people simply inconceivable that there exist numbers less than nothing.

One of the disturbing arguments used by Bernoulli was: it must be true that $\ln(x) = \ln(-x)$ for all $x \neq 0$, because

$$f(x) = \ln(x) \implies f'(x) = \frac{1}{x} \quad \text{and}$$
$$g(x) = \ln(-x) \implies g'(x) = \frac{(-1)}{-x} = \frac{1}{x} \quad \text{(by the chain rule)} \qquad \text{(A)}$$

Note that we immediately run into troubles at this point because by our standard rules (7.1) we should have:

$$\ln(-x) = \ln[(-1)x] = \ln(-1) + \ln(x),$$

but this would imply that $\ln(-1) = 0$ which Bernoulli knew could not be true.

Leibniz objected that logarithms of negative numbers must be *imaginary*. When the correspondence between Leibniz and Bernoulli became published this acted as a tremendous stimulus on Leonhard Euler, who was Bernoulli's student. In two epoch-making papers 1747 and 1749 Euler carefully worked out this problem and found that its solution lies at an unexpected place: *all numbers except zero have an infinity of logarithms*. His proof is based on one of the most remarkable formulas in mathematical analysis, Euler's formula, as it is called today:

Leonhard Euler
1707–1783

$$e^{ix} = \cos(x) + i\sin(x), \qquad (7.12)$$

where i denotes the *imaginary unit*, defined[7] by $i^2 = -1$. Since the sine and cosine functions are periodic with period 2π, it follows that

$$\ln(\cos(x) + i\sin(x)) = i(x \pm 2n\pi), n \in \mathbb{N} \qquad (7.13)$$

Setting $x = \pi$, we obtain $\cos(\pi) + i\sin(\pi) = -1$ and

$$\ln(-1) = \pm\pi i, \quad \pm 3\pi i, \ldots,$$

and none of these values is zero. That $\ln(x)$ is *multivalued* causes the problem that $\ln(x)$ is no longer a function in the strict sense, therefore some additional restrictions are necessary. This leads to the concept of the *principal value* of the logarithm which is defined such that the *argument* ϕ or *angle* of a complex number $z = |z|e^{\phi i}$ is restricted to the interval $-\pi < \phi \leq \pi$. As a consequence our standard rules (7.1) do not always hold. For instance, it is not generally

[7] The symbol i was introduced by Euler himself, although in the aforementioned papers of 1747 and 1749 he mostly writes $\sqrt{-1}$ instead of i.

true that $\ln(a \cdot b) = \ln(a) + \ln(b)$. A counter example directly following from (7.12) and (7.12) is this one:

$$\ln(-i) = \ln((-1) \cdot i) = \ln(-1) + \ln(i) = \pi i + \frac{\pi i}{2} = \frac{3\pi i}{2},$$

but

$$\ln(-i) = -\frac{\pi}{2} \neq \frac{3\pi}{2},$$

which follows from (7.13).

This is a dangerous trap when you are performing numerical calculations with complex logarithms, as computer software generally uses principal values.

If you want to elaborate on these interesting aspects of logarithms then you should definitely read the original papers of Euler (see the annotated bibliography in Section 3 below). The historical controversy between Leibniz and Bernoulli is discussed in Cajori (1913b) where you can find a synopsis of this famous correspondence. Cajori (1913c) is devoted to Euler's contributions and includes a synopsis of the correspondence on this subject between Euler and D'Alembert from April 15, 1747 to September 28, 1748.

7.3 An Annotated Bibliography

There is an enormous amount of literature on the history of logarithms. In several books on the history of mathematics in general you will find detailed accounts of John Napier, Henry Briggs and their time, including also developments like decimal fractions and the invention of modern mathematical notation. In the sequel I want to draw your attention to a few books which I found very interesting.

Boyer and Merzbach (2011) is a very readable and rather complete textbook on the history of mathematics. This is also true of Struik (2008), a book first published in 1948. Also recommendable is Wussing (2008). Sonar (2016) is devoted primarily to the famous Newton-Leibniz Controversy, but it sheds also some light on other developments in mathematics during the 17th century. For instance the discovery of the Mercator series is described in detail.

Papers on logarithms and their history continue to be published since the 19th century, often when there is an anniversary. Florian Cajori is author of an outstanding series of papers. In Cajori (1913a) you find an account of the work of Napier and Briggs, Cajori (1913b) and Cajori (1913c) are devoted to the early discussions of logarithms of negative and imaginary values. In Cajori (1913d) you find an exposition of the developments up to 1800, which is interesting insofar as Euler's 1747 paper was not published before 1862. Logarithms viewed as complex functions and the idea of a principal value are presented in Cajori (1913e) and Cajori (1913f).

Also, you *should read* the fine overviews due to Panagiotou (2011) and Burn (2001) which cover in detail the invention of hyperbolic or natural logarithms, a

development which foreshadowed the revolutionary discoveries of the differential and integral calculus. Also recommended is the profound study by Denis Roegel (2010). It is quite voluminous as it contains a reconstruction of Briggs' tables. But on the first 34 pages you find a detailed elaboration of the techniques used by Briggs to calculate common logarithms.

Raymond Ayoub's (1993) paper is a rather complete and very readable exposition of the mathematics of Naperian logarithms.

And last but not least, please read the excellent papers by Leonhard Euler: Euler (1747) and Euler (1749). For both papers English translations of the original french text (thanks to Stacy Langton and Todd Doucet) are available from the *Euler Archive* (`http://eulerarchive.maa.org/`).

7.4 References

[1] Raymond Ayoub. "What is a Naperian Logarithm?" In: *The American Mathematical Monthly* 100.4 (1993), pp. 351–364.

[2] Carl B. Boyer and Uta C. Merzbach. *A History of Mathematics*. John Wiley & Sons, 2011.

[3] R. P. Burn. "Alphonse Antonio de Sarasa and Logarithms". In: *Historia Mathematica* 28 (2001), pp. 1–17.

[4] Florian Cajori. "History of the Exponential and Logarithmic Concepts". In: *The American mathematical Monthely* 20.1 (1913), pp. 5–14.

[5] Florian Cajori. "History of the Exponential and Logarithmic Concepts". In: *The American mathematical Monthely* 20.2 (1913), pp. 35–47.

[6] Florian Cajori. "History of the Exponential and Logarithmic Concepts". In: *The American mathematical Monthely* 20.3 (1913), pp. 75–84.

[7] Florian Cajori. "History of the Exponential and Logarithmic Concepts". In: *The American mathematical Monthely* 20.4 (1913), pp. 107–117.

[8] Florian Cajori. "History of the Exponential and Logarithmic Concepts". In: *The American mathematical Monthely* 20.5 (1913), pp. 148–151.

[9] Florian Cajori. "History of the Exponential and Logarithmic Concepts". In: *The American mathematical Monthely* 20.7 (1913), pp. 205–210.

[10] B. C. Carlson. "An Algorithm for Computing Logarithms and Arctangents". In: *Mathematics of Computation* 26.118 (1972), pp. 543–549.

[11] Leonhard Euler. *De la controverse entre Mrs. Leibniz et Bernoulli sur les logarithmes des nombres negatifs et imaginaires*. 1747. URL: http://eulerarchive.maa.org/docs/translations/E168en.pdf.

[12] Leonhard Euler. *Sur les logarithmes des nombres negativs et imaginaires*. 1749. URL: http://eulerarchive.maa.org/docs/translations/E807en.pdf.

[13] W. B. Jones and W. J. Thron. *Continued Fractions - Analytic Theory and Applications*. Reading, MA, USA: Addison-Wesley, 1980.

[14] Morris Kline. *Mathematics - The Loss of Certainty*. Oxford University Press, 1980.

[15] E. N. Panagiotou. "Using History to Teach Mathematics: The Case of Logarithms". In: *Science & Education* 20 (2011), pp. 1–35.

[16] R. C. Pierce. "A Brief History of Logarithms". In: *The Two-Year College Mathematics Journal* 8 (1977), pp. 22–26.

[17] Denis Roegel. *A reconstruction of the tables of Briggs' Arithmetica logarithmica (1624)*. 2010. URL: http://locomat.loria.fr/briggs1624/briggs1624doc.pdf.

[18] Thomas Sonar. *Die Geschichte des Prioritätenstreits zwischen Leibniz und Newton*. Springer Spektrum, 2016.

[19] Dirk J. Struik. *A Concise History of Mathematics*. 4th. Dover Publications, 2008.

[20] A. J. Thompson and Karl Pearson. "Henry Briggs and His Work on Logarithms". In: *The American Mathematical Monthly* 32.3 (1925), pp. 129–131.

[21] Hans Wussing. *6000 Jahre Mathematik - Eine kulturgeschichtliche Zeitreise*. Vol. 1, Von den Anfängen bis Leibniz und Newton. Springer Verlag, 2008.

Topic 8

Exercise Number One
Partition Theory

Keywords: *partitions of integers, the Money Changing Problem, combinatorics, generating functions*

This chapter has not been finished yet.
October 4, 2018

8.1 An Invitation

8.1.1 Exercise number one

This topic should introduce you into one of the most fascinating areas of discrete mathematics: the theory of integer partitions.

To begin with, let us have a look at this famous problem:

1. *Auf wieviel Arten läßt sich ein Franken in Kleingeld umwechseln? Als Kleingeld kommen (in der Schweiz) in Betracht: 1-, 2-, 5-, 10- ,20- und 50 Rappenstücke (1 Franken = 100 Rappen).*

Can you find the answer?

This is *Exercise 1* in *Aufgaben und Lehrsätze aus der Analysis I* by George Pólya and Gábor Szegő, one of the classical textbooks in mathematics, the first edition published in 1925. It is also known as *Money Changing Problem* and has been discussed and solved already by Leonhard Euler in the 18th century.

8.1.2 Partitions of integers

Technically speaking solving *Exercise 1* requires to count a special class of *integer partitions*.

A partition of a positive integer n is a sequence of integers $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_k)$, such that

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0 \tag{A}$$


George Pólya (1887–1985) and Gábor Szegő (1895–1985)

and

$$\lambda_1 + \lambda_2 + \ldots + \lambda_k = n.$$

The numbers λ_i are called the *parts* of $\boldsymbol{\lambda}$. Symbolically one writes $n \dashv \boldsymbol{\lambda}$ or $\boldsymbol{\lambda} \vdash n$ to express the fact that n is split into parts given by $\boldsymbol{\lambda}$.

The *partition function* $p(n)$ counts the number of partitions of n. For instance, $\boldsymbol{\lambda} = (2, 1, 1, 1) \vdash 5$, because $5 = 2 + 1 + 1 + 1$. This $\boldsymbol{\lambda}$ is only one out of seven partitions of 5, indeed, we have

$$\begin{aligned} 5 &= 5 \\ &= 4 + 1 \\ &= 3 + 2 \\ &= 3 + 1 + 1 \\ &= 2 + 2 + 1 \\ &= 2 + 1 + 1 + 1 \\ &= 1 + 1 + 1 + 1 + 1, \end{aligned}$$

thus $p(5) = 7$. Note, that due to condition (A) the *order* of summands is not taken into account, i.e. the partitions $\boldsymbol{\lambda}_1 = (3, 2)$ and $\boldsymbol{\lambda}_2 = (2, 3)$ are considered to be the same object.

The partition function $p(n)$ grows very fast. Indeed, it can be shown that

$$\begin{gathered} p(10) = 42, \quad p(20) = 627, \quad p(50) = 204226, \\ p(100) = 190569292, \quad p(200) = 3972999029388, \quad \text{etc.} \end{gathered}$$

Is there a formula for $p(n)$? Yes, and this is one of the most exciting results of 20th-century mathemtics, the celebrated *Hardy-Ramanujan-Rademacher Formula*. Unfortunately, this formula is extremely complicated, but it yields a simple approximation:

$$p(n) \sim \frac{1}{4n\sqrt{3}} \exp\left[\pi\sqrt{2n/3}\right] \quad \text{for} \quad n \to \infty$$

In 1929 J. E. Littlewood has written a fascinating review of the *Collected Papers of Srinivasa Ramanujan* which you should read in order to get an impression of the ingenuity of S. Ramanujan, see the references below.

Are there other ways to calculate $p(n)$ exactly? Find it out! Actually, there are several alternatives to the Hardy-Ramanujan-Rademacher Formula.

8.1.3 Partitions with restricted parts

Very often one is interested in partitions of n such that the parts λ_i satisfy certain conditions. For instance, we may consider partitions with all parts being odd integers. Taking $n = 5$, then the partitions of 5 into odd parts are:

$$\begin{aligned} 5 &= 5 \\ &= 3 + 1 + 1 \\ &= 1 + 1 + 1 + 1 + 1, \end{aligned}$$

and

$$p(n|\text{all parts odd}) = 3 \tag{B}$$

We may also consider partitions such that all parts are different. For $n = 5$ we find:

$$\begin{aligned} 5 &= 5 \\ &= 4 + 1 \\ &= 3 + 2, \end{aligned}$$

and

$$p(n|\text{all parts different}) = 3 \tag{C}$$

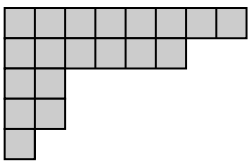
Interestingly, (B) and (C) yield the same value for $n = 5$. A mere coincidence, or is there a general rule?

Indeed, Euler already discovered and proved that

$$p(n|\text{all parts odd}) = p(n|\text{all parts different}) \tag{D}$$

for all $n \geq 0$. This is one, presumably the most famous, of many known so-called *partition identities*. How to prove (D)? This can be done in several ways. You should discuss at least one of them.

A simple and very powerful device is a graphical representation of a partition called *Young diagram*. It consists of rows of boxes, like



which represents the partition $19 \dashv (8, 6, 2, 2, 1)$. A very similar representation are *Ferrers graphs*, which use circles instead of boxes.

Several partition identities are directly deducible from these graphical devices. Here is an example of a partition identity which is easily proved using Young diagrams:

The number of partitions of n with largest part equal to k equals the number of partitions of n with exactly k parts.

Prove it!

The idea of partitions into odd parts can be generalized. Let $\mathcal{A}$ by a finite or infinite subset of the set of positive integers $\mathbb{N}$. For instance, $\mathcal{A} = \{1, 3, 5, \ldots\}$, the set of odd numbers. Then a very interesting problem is to find the number of partitions $\boldsymbol{\lambda} \vdash n$ such that all parts are elements of $\mathcal{A}$, in other words, what is

$$p(n|\text{all } \lambda_i \in \mathcal{A}) \quad ? \tag{E}$$

Observe, when

$$\mathcal{A} = \{1, 2, 5, 10, 20, 50\},$$

then (E) is the answer to the Money Changing Problem of *Exercise 1.*

But beware!

Consider a country with only 9, 17, 31 and 1000 € bills. How many ways are there to change a 1000 € bill?

It is not all clear that there is even one possiblity to give such a change! In fact, the problem of determining if even one solution exists is known to by very hard, indeed, it is NP-hard.

8.1.4 Generating functions

The most important tool in the study of partitions are *generating functions.* They may be viewed as some sort of clotheslines on which we hang up numbers we want to display, e.g. the partition numbers $p(n)$. Technically, generating functions are *power series* in some variable z, e.g. the generating function $P(z)$ of the partition numbers $p(n)$ is

$$\begin{aligned} P(z) &= p(0) + p(1)z + p(2)z + p(3)z + \ldots \\ &= 1 + z + 2z^2 + 3z^3 + 5z^4 + 7z^5 + 11z^6 + \ldots \end{aligned}$$

It is remarkable that this function can be written as an infinite product:

$$P(z) = \frac{1}{1-z}\,\frac{1}{1-z^2}\,\frac{1}{1-z^3}\cdots = \prod_{i=1}^{\infty} \frac{1}{1-z^i} \tag{F}$$

This is a really fundamental relation discovered by Leonhard Euler. Two of the most important points about generating functions are:

- If we have a function term $P(z)$ for the generating function like (F), then we may be able to extract the coefficient of z^n in $P(z)$ by various techniques. Indeed, *all information about the counting sequence $p(n)$ is contained in $P(z)$*!
- Furthermore, having understood the genesis of (F), you will be able to find generating functions for various restricted partition numbers. As an example, the generating function of the number of partitions with all parts different can be shown to be

$$P(z|\text{all parts different}) = (1+z)(1+z^2)(1+z^3)\cdots = \prod_{i=1}^{\infty}(1+z^i)$$

  There are many other examples. Particularly interesting is the generating function of the number of partitions with parts taken from some finite or infinite set $\mathcal{A}$. Recall the Money Changing Problem!

8.2 Where to go from here

8.2.1 Issues of general interest

- Prepare a careful though interesting introduction into partition theory.
- Discuss some partition identities and prove them. There are two major techniques of proof: the method of bijections and generating functions, compare these methods.
- Solve the Money Changing Problem as presented in *Exercise 1.* of Pólya and Szegő.
- Partition theory has many applications, e.g., in computer science but also in statistics. A prominent application is the Wilcoxon rank sum test, or equivalently Mann-Whitney's U-statistic. Show how this famous 2-sample test is related to partitions.

8.2.2 Some more suggestions

- Describe and implement an algorithm to generate *systematically* all partitions of n, of course n must not be too large. You may do this in R, `octave/matlab`, whatever you want. How to handle restricted partitions? It may be interesting also to experiment with some symbolic compution software like *Mathematica* or *Maple*. As far as I know, *Mathematica* is accesible to you via a WU-campus license.

8.3 An Annotated Bibliography

The literature on integer partitions is enormous. Here are a few important resources.

- You should start your reading with Andrews and Eriksson (2004). This booklet, available as paperback, as an excellently written introductory textbook.
- George Andrews is the *grand seigneur* of partition theory. Andrews (2003) is a classical text written for professional mathematicians. This book it is a reprint of the 1976 edition which has been published originally as part of the *Encyclopedia of Mathematics*. However, chapters 1 and 2 are easy to read and so it may be profitable for you to have a look at these. Chapter 5 gives a thorough derivation and proof of the *Hardy-Ramanujan-Rademacher Formula*. Chapter 14 is particularly useful, if you want to develop algorithms for systematically counting various types of partitions.
- Sedgewick and Flajolet (2009) is a wonderful textbook. Partitions are not treated systematically in this book, but there is a wealth of material on partitions spread over the book. The first chapters introduce the so-called *symbolic method*, an extremely powerful and elegant technique to find generating functions.
- The famous book by Pólya and Szegő has also been translated into English. Exercises 1-27 are related to partition problems. You may wonder why the Money Changing Problem, which is obviously of combinatorial nature, has found its place in a book on analysis?
- Regarding Wilf (1990a): the title is program! A free pdf-version of this book is available, see Wilf (1990b).

More is still missing ...

8.4 References

[1] G. E. Andrews. *The Theory of Partitions*. Cambridge University Press, 2003.

[2] G. E. Andrews and K. Eriksson. *Integer Partitions*. Cambridge University Press, 2004.

[3] J. E Littlewood. "Collected Papers of Srinivasa Ramanujan". In: *Mathematical Gazette* 14 (1929), pp. 427–428.

[4] Robert Sedgewick and Philippe Flajolet. *Analytic Combinatorics*. Cambridge University Press, 2009.

[5] H. S. Wilf. *Generatingfunctionology*. Academic Press, 1990.

[6] H. S. Wilf. *Generatingfunctionology*. 1990. URL: http://www.math.upenn.edu/~wilf/gfology2.pdf.

Topic 9

The Ubiquitious Binomialcoefficient

Keywords: *discrete mathematics, binomial theorem, binomial identities, summation of series*

9.1 An Invitation

9.1.1 The classical binomialtheorem

From elementary mathematics you are certainly familiar with the classical binomial coefficient

$$\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}, \tag{9.1}$$

where n and k are nonnegative integers and $k!$ denotes the well-known factorial function $k! = k \cdot (k-1) \cdots 2 \cdot 1$. The symbol $\binom{n}{k}$ is usually read as n *choose* k, n is called the *upper index*, k the *lower index*. It is very likely that you are also aware of

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, \tag{9.2}$$

a representation of binomial coefficients which is very common, though from a computational point of view not the best we can have. Binomial coefficients as we have defined them so far are always nonnegative integers. This is by no means clear apriori if you look at (1) or (2).

The name binomial coefficient stems from the fact that these numbers occur as coefficients in the expansion of the binomial $(1+z)^n$ into ascending powers of z, viz:

$$(1+z)^n = \binom{n}{0} + \binom{n}{1}z + \binom{n}{2}z^2 + \ldots + \binom{n}{n-1}z^{n-1} + \binom{n}{n}z^n \tag{9.3}$$

This formula is known as the *(classical) Binomial Theorem*, and the binomial function $f(z) = (1+z)^n$ is also called the *generating function* of the binomial coefficients, a very important concept in mathematics. You should check that

$$\binom{n}{k} = 0, \qquad \text{for all } k > n,$$

and therefore the series (9.3) is always terminating, indeed, it is a polynomial of degree n in z.

Using formula (1) which is due to Blaise Pascal (1623-1662), we find successively:

$$\begin{array}{lcl} (1+z) & = & 1+z \\ (1+z)^2 & = & 1+2z+z^2 \\ (1+z)^3 & = & 1+3z+3z^2+z^3 \\ (1+z)^4 & = & 1+4z+6z^2+4z^3+z^4 \\ \dots & & \dots \end{array}$$

The first three expansions have been known already in ancient times, e.g. they where known to Euklid (around 300 BC) and Diophantus (215–299?).

Pascal's formula can easily be found using a simple combinatorial argument. Just rewrite:

$$(1+z)^n = \underbrace{(1+z)(1+z)\cdots(1+z)}_{n \text{ factors}}, \tag{9.4}$$

and now find out how in this n-fold product the term x^k is composed. You should work out this argument in precise terms in your thesis and thereby show that $\binom{n}{k}$ equals the number of ways to form subsets of size k out of a groundset having n elements.

9.1.2 Pascal's triangle

The binomial coefficients can be arranged in a triangular array. The first lines of this array read as:

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\binom{n}{8}$	$\binom{n}{9}$	$\binom{n}{10}$
0	1										
1	1	1									
2	1	2	1								
3	1	3	3	1							
4	1	4	6	4	1						
5	1	5	10	10	5	1					
6	1	6	15	20	15	6	1				
7	1	7	21	35	35	21	7	1			
8	1	8	28	56	70	56	28	8	1		
9	1	9	36	84	126	126	84	36	9	1	
10	1	10	45	120	210	252	210	120	45	10	1

This array is commonly known as *Pascal's Triangle*, but it was known long before Pascal, e.g. it appears in papers of the chinese mathematician Chu-Shih-Chieh around 1300. About Chu we will have to say more in a few moments. By the way, you can find a lot of interesting historical information about the binomial theorem in Coolidge (1949).

Pascal's Triangle has many remarkable properties. Here are a few observations:

- The rows of the triangle are *unimodal*, this means that the numbers in any row first increase up to a maximum value located in the middle and then they decrease again. Unimodality of binomial coefficients is the result of an important *symmetry property*:

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

- The rows are recognized as sequence number A007318 in Sloane' On-Line Encyclopedia of Integer Sequences (`https://oeis.org/`), a fascinating and very useful web-site.
- The central term in row n increases very fast as n increases.
- Several recurrence relations between entries in different rows can be identified.

Let us now comment briefly on the last two observations. For the first one, assume that n is an *even* number, so $n = 2m$. Then there is a unique central term in row $n = 2m$, $\binom{2m}{m}$. It can be shown that

$$\binom{2m}{m} \sim \frac{2^{2m}}{\sqrt{m\pi}}, \qquad m \to \infty \tag{9.5}$$

In this formula the symbol $\sim$ means that the ratio of the left and the right side of (9.5) tends to 1 as $m \to \infty$.

Blaise Pascal (1623-1662)

This important *asymptotic formula* tells us that $\binom{2m}{m}$ grows roughly as fast as 2^{2m} only slowed down slightly by the factor $1/\sqrt{m\pi}$ [1]. In other words, we have almost exponential growth. Formula (9.5) can be proved using the celebrated *Stirling Formula*. You should discuss this approximation formula in your thesis and also provide a proof. That can be done by more or less elementary methods.

Regarding recurrence relations between various entries of the triangle: here is the most famous one, it is indeed on place four of the *Top ten binomial coefficient identities*, see (Graham, Knuth, and Patashnik, 2003, p.171):

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}. \tag{9.6}$$

Let's give it a try: for $n = 10$ and $k = 4$ formula (6) states that

$$\binom{10}{4} = \binom{9}{4} + \binom{9}{3}.$$

Using the table above, we have indeed: $210 = 126 + 84$.

How can we prove (6)?

There are several ways to prove (6). The easiest way is to use *mathematical induction*. In your thesis you should explain the *induction principle* and show

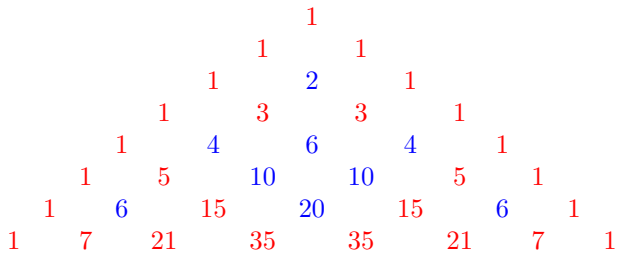
[1] The occurrence of π in this formula is somewhat a miracle.

by example how it works. One such example must be (6), another one formula (1).

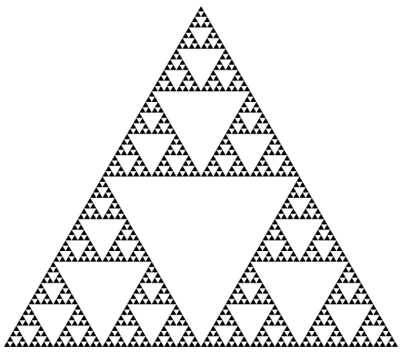
Many interesting number theoretic properties of binomial coefficients are buried in Pascal's Triangle.

Here is one: if n and k are relatively prime, which means that their greatest common divisor equals 1, then $\binom{n}{k}$ is divisible by n. For instance, the greatest common divisor of $n = 8$ and $k = 3$ equals 1, and $\binom{8}{3} = 56$ is indeed divisible by 8. As a special case of this statement we have: for any prime number p and any k such that $0 < k < p$ the binomial coefficient $\binom{p}{k}$ is divisible by p. Can you prove these statements?

And one more exciting property: let us rewrite Pascal's Triangle in the following more or less standard form and draw odd entries in red, even entries in blue:



The pattern showing up resembles a *Sierpinski Triangle*, a famous *fractal structure*:



9.1.3 Newton's binomial theorem

Let us now return to the classical binomial theorem (3). Arround 1664 or 1665 Newton considered the question: what happens, if the exponent n is not a nonnegative integer?

This question leads us to consider the expansion of $(1+z)^\alpha$, where the exponent α may be any real number. Newton used the idea of *analogy*, one of his favorite principles:

$$(1+z)^\alpha = \binom{\alpha}{0} + \binom{\alpha}{1} z + \binom{\alpha}{2} z^2 + \binom{\alpha}{3} z^3 + \dots \tag{9.7}$$

But what meaning should we give to $\binom{\alpha}{k}$? Newton argued that the definition (1) of binomial coefficients continues to hold. Indeed, in (1) we do not really require that n is a nonnegative integer. If we rewrite (1) with n replaced by α, then we have:

$$\binom{\alpha}{k} = \frac{\alpha(\alpha-1)(\alpha-2)\cdots(\alpha-k+1)}{k!}. \tag{9.8}$$

A closer look at (8) reveals that $\binom{\alpha}{k}$ is a polynomial of degree k in α. So, α may be *any real number*! But $\binom{\alpha}{k}$ no longer becomes zero when $k > \alpha$ unless α is a nonnegative integer. This observation has an important consequence: the expansion (9.7) is no longer a polynomial in z, it is, in general, an *infinite series*. At this point *convergence* becomes an issue. It can be shown that (9.7) converges if and only if z is sufficiently small, more precisely we require $|z| < 1$.

Let's give it a try and put $\alpha = -1$. Then by (9.8):

$$\binom{-1}{k} = \frac{(-1)(-2)(-3)\cdots(-k)}{k!} = \frac{(-1)^k k!}{k!} = (-1)^k,$$

so

$$(1+z)^{-1} = \frac{1}{1+z} = 1 - z + z^2 - z^3 + z^4 - \ldots,$$

a well-known variant of the *infinite geometric series*. Putting $z \to -z$ above yields:

$$(1-z)^{-1} = \frac{1}{1-z} = 1 + z + z^2 + z^3 + z^4 + \ldots.$$

More generally we may consider expansions like

$$(1+z)^{-n} = \binom{-n}{0} + \binom{-n}{0} z + \binom{-n}{2} z^2 + \ldots,$$

where n is a nonnegative integer.

Let us *negate the upper index* in each of these binomial coefficients, which is done as follows:

$$\begin{aligned}\binom{-n}{k} &= \frac{(-n)(-n-1)\cdots(-n-k+1)}{k!} \\ &= (-1)^k \frac{n(n+1)(n+2)\cdots(n+k-1)}{k!} = (-1)^k \binom{n+k-1}{k},\end{aligned}$$

by (1). Here we see one more remarkable and important relation. As a consequence we have the alternative expansion:

$$(1+z)^{-n} = \binom{n-1}{0} - \binom{n}{1} z - \binom{n+1}{2} z^2 - \ldots = \sum_{k\geq 0} (-1)^k \binom{n+k-1}{k} z^k$$

For instance, if we set $n = 3$, then we get (verify please!):

$$(1+z)^{-3} = \frac{1}{(1+z)^3} = 1 - 3z + 6z^2 - 10z^3 + 15z^4 - \dots$$

Now a more exciting case: consider $\alpha = 1/2$, then

$$(1+z)^{1/2} = \sqrt{1+z} = \binom{1/2}{0} + \binom{1/2}{1} z + \binom{1/2}{2} z^2 + \dots \tag{9.9}$$

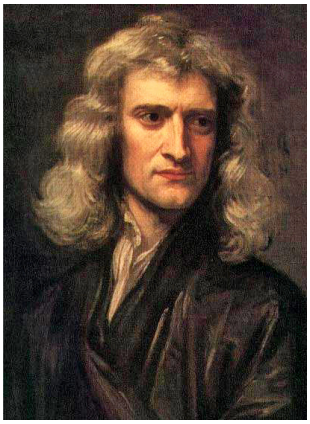
Again, let us rewrite the binomial coefficients occuring in this expansion using (9.8). You will find (please verify):

$$\binom{1/2}{k} = \frac{(-1)^{k-1}}{2^{2k-1}} \cdot \frac{1}{k} \binom{2k-2}{k-1}, \quad \text{for } k > 0. \tag{9.10}$$

This results in an entirely different series expansion for the square root:

$$\sqrt{1+z} = 1 - \frac{1}{2} \sum_{k \geq 1} \frac{1}{k} \binom{2k-2}{k-1} \left(-\frac{z}{4}\right)^k$$

The transformation sketched above is also known as *going halves*. There are many other such transformations.



Isaac Newton
(1642-1727)

Some historical remarks are in order: (9.7) is commonly referred to as *Newton's Binomial Theorem*. Newton communicated his ideas in two letters written 1676 to Henry Oldenburg, secretary of the Royal Society. Actually, the first of these letters has been addressed originally to G. Leibniz, but by incidence got delivered to Oldenburg. Interestingly, Newton did not elaborate (9.7) in full generality, he only considered some special cases and he did not discuss the problem of convergence, see Boyer and Merzbach (2011). A complete proof of Newton's binomial theorem was not given before Abel (1826). A scan of this paper is available in the web (see the references below). The cited issue of Crelle's Journal contains six (!) papers of Abel, among them his nowadays classical proof of the impossibility of solving polynomial equations of order higher than four by radicals. This is quite remarkable, as Abel was at that time only 24 years old. Unfortunately, he died three years later.

9.1.4 Binomial sums

And now we are coming to the really thrilling part of the story, *binomial sums*. These sums involve one or more binomial coefficients, they appear in practically all areas of mathematics and have been subject to thorough investigation over centuries.

Let us begin with two harmless examples (n is in both a nonnegative integer).

$$\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{n}=2^n$$
$$\binom{n}{0}-\binom{n}{1}+\binom{n}{2}-\ldots+(-1)^n\binom{n}{n}=0$$

What do you think about these binomial sums? How can we prove them? One way is to use the principle of induction. But there is a much easier way. Just put $z=1$ in (9.3) to get the first of these sums, then put $z=-1$ in (9.3) to obtain the second sum. Note that we need not worry about convergence, as n is assumed to be nonnegative integral, so (9.3) is always terminating. We could not have used this trick in (9.7). Thus it *not true* that

$$\sqrt{2}=\binom{1/2}{0}+\binom{1/2}{1}+\binom{1/2}{2}+\ldots$$

which would result when we put $z=1$ in (9.9).

Here is another famous sum:

$$\sum_{k\geq 0}\binom{n}{k}\binom{m}{a-k}=\binom{n+m}{a}. \tag{9.11}$$

(9.11) is known as *Chu - Vandermonde Formula*. It has an enormous number of applications. You should derive and prove it in your thesis.

Many ideas and methods have been developed to evaluate binomial sums. Quite for a long time much of this work had the character of a case-by-case analysis, a typical compendium is the book of Riordan (1968). Things have changed, however, since the publication of a famous paper by George George E. Andrews (1974, Section 5).

Today, we have a wonderful and deep theory of such sums, so that many of them (though by no means all) can be evaluated in a rather routine manner. When preparing your thesis you should learn to handle at least in part some of these methods, so that you will be able to evaluate for instance this miraculous sum:

$$\sum_{k\geq 0}\binom{n+k}{2k}\binom{2k}{k}\frac{(-1)^k}{k+1}$$

You will be surprised!

9.2 Where to go from here

- Give an interesting and readable overview of binomial coefficients.
- Your thesis should contain a discussion of the binomial theorem, the classical one and Newton's theorem. Regarding the latter, you will also have to use *Taylor's Formula* which you can find in any textbook on elementary differential calculus.

- Discuss important properties of binomial coefficients, these may also include some remarkable number theoretic properties.
- Present a collection of identities between binomial coefficients and provide proofs. In this context you should explain the *Principle of Induction*.
- Discuss various transformations like *going halves*, *negating the upper index*, etc. Show how these can be used to simplify binomial sums. Give some examples of summations.

Note. Your own ideas and creativity are always welcome!

9.3 An Annotated Bibliography

The book Graham, Knuth, and Patashnik (2003) is certainly the most important and helpful one, in particular Chapter 5 (about 100 pages) contains a lot of material presented in really excellent form.

Regarding number theoretic properties of binomial coefficients the book of G. E. Andrews (1994) is an easy-to-read introduction to the theory of numbers which is very helpful e.g. if you want to learn about congruences. This book is also available for free download.

Abel's original paper is available e.g. at Göttinger Digitalisierungszentrum (`http://gdz.sub.uni-goettingen.de/gdz/`).

9.4 References

[1] Niels Henrik Abel. "Untersuchungen über die Reihe $1+\frac{m}{1}x+\frac{m(m-1)}{2}x^2\ldots$". In: *Crelle's Journal für die reine und angewandte Mathematik* 1 (1826), pp. 311–366.

[2] G. E. Andrews. *Number Theory*. Dover Books on Mathematics. Dover Publications, 1994.

[3] George E. Andrews. "Applications of Basic Hypergeometric Functions". In: *SIAM Review* 16.4 (1974), pp. 441–484.

[4] Carl B. Boyer and Uta C. Merzbach. *A History of Mathematics*. John Wiley & Sons, 2011.

[5] J. L. Coolidge. "The story of the binomial theorem". In: *The American Mathematical Monthly* 56 (1949), pp. 147–157.

[6] L. Graham Ronald, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. 2nd ed. Addison-Wesly, 2003.

[7] J. Riordan. *Combinatorial Identities*. Wiley series in probability and mathematical statistics. Wiley, 1968.

Topic 10

Prime Time for a Prime Number

Keywords: *elementary number theory, computational number theory, prime numbers, modular arithmetic, public key encryption, internet data security*

10.1 An Invitation

10.1.1 A new world record

On 20 January 2016, BBC News headlined: *Largest known prime number discovered in Missouri*! Immediately many other TV-stations and newspapers followed and posted similar messages. For instance, the New York Times on 21 January: *New Biggest Prime Number = 2 to the 74 Mil ... Uh, It's Big*!

What is this beast, let us call it *bigP*, that received such a wide media response? Well, here it is:

$$bigP = 2^{24\,207\,281} - 1$$

In decimal notation this number has 22 338 618 digits. It has been found on January 7 by a team of the Great Internet Mersenne Prime Search Project (GIMPS).

10.1.2 Why primes are interesting

Ok, people tell us that *bigP* is a prime number. But what is a prime?

May be, you recall the definition of a prime number. For definiteness here it is: *A natural number p is prime, if it has exactly two distinct divisors,* 1 *and* p. The list of primes starts with

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, \ldots$$

A number[1] that is not prime is called *composite*.

[1] In the sequel when the term *number* occurs, it always means a natural number.

Note that 1 is not considered a prime. There are very good reasons for excluding 1 from the set of primes. Stated in simple terms, if we let 1 be a prime, most non-trivial theorems about primes would be rendered false. So, 1 is not a prime.

Probably you also know that there a infinitely many primes, and you may have heard that primes are in a certain sense the building blocks of all natural numbers because any number can be *factored* into primes. This is the statement of the *Fundamental Theorem of Arithmetic*. E.g.,

$$15624 = 2^3 \cdot 3^2 \cdot 7 \cdot 31$$

So, primes are very special numbers.

You may have already guessed it: this thesis should be about prime numbers. More precisely, it should be about the *elementary* theory of prime numbers and it should also be a coverage of interesting computational issues related to primes.

At this point you may stop reading further this description and argue:

1. I have never learned anything about primes except for some really basic facts, neither at high school nor during my university studies at WU.
2. So, to master this thesis, I will have to learn quite a lot about number theory.
3. Frankly speaking, number theory is quite an esoteric part of mathematics, only real nerds are working in this field. And most importantly, as I am studying economics at WU, when I take a thesis topic with a mathematical flavor, then it must be *useful mathematics*. But number theory and especially prime numbers seem to be pretty useless stuff!

Let me briefly comment on these objections:

Ad (1): True.

Ad (2): Very true, *but see below*.

Ad (3): Completely wrong! Give me a chance to explain why.

10.1.3 Primes and RSA-encryption

Very likely you are customer of one or the other internet trading site, e.g. Amazon. After having made your choice and put some stuff into your virtual shopping basket you will have to pay. Usually you will have to enter your credit card number or other sensible information about your bank account. And if you do so, don't you worry that this information may be stolen, may fall into the wrong hands? After all, sending a message over the internet is no more secure then sending a postcard. In principle, everybody can read it.

Of course, the solution is to *encrypt* sensible information. For this purpose modern web browsers communicate with web servers using a special protocol

called *Secure Socket Layer*. Whenever SSL is used you can see this in the command line of your browser. The web address you are communicating with is preceded by the word `https`. Data sent over SSL is encrypted by some encryption system and the receiver of the secret message deciphers it. But there is a problem. Deciphering an encoded messages requires a *key*. Until the 1970s it was necessary to *exchange* a key, e.g. during a secret meeting. But did you ever meet an agent of Amazon at a clandestine location and fix a key word, say `yellowsubmarine`?

No, you won't do that, because:

- This procedure is potentially unsafe for several reasons. Exchanging the key may be intercepted, but more importantly, using the same key word for a longer time makes it very likely that your code will be broken.
- After all, this procedure is totally impractical! No system of e-commerce or e-banking with many thousands of customers would be possible on this basis.

But in the early 1970s *public-key cryptography* has been invented and now prime numbers come into play!

This goes as follows (basically the so-called RSA cryptosystem):

- The security server at Amazon generates two large primes p and q. Here large means some several hundred decimal digits.
- The server then multiplies p and q to give a large composite number $m = p \cdot q$.
- The number m is sent to the web client, e.g. your browser. This uses m to encrypt sensible information and sends this back to the security server.
- Amazon's server decrypts the secret message and now knows about your account information, credit card number, etc.

How does this methods work? Why is it considered secure? The crucial point is:

Given a large composite number m, finding its prime factors is an extremely difficult task. Until now no algorithms are known which can do that job in reasonable time on computer hardware currently available for civil purposes.

Thus Amazon can send the number m to your server, your client browser sends back an encrpyted message using m as key, say $t(m)$, and although the unauthorized eavesdropper can read m and the encrypted message $t(m)$ he will not be able to decipher $t(m)$ in reasonable time because finding factors of large numbers is so difficult.

But how can Amazon decipher $t(m)$? This is possible because number theory provides so marvelous tools like *modular arithmetic* and *Fermat's Little Theorem* for us. We shall have to say more about these in a few minutes.

Let us pause for a moment here. If you have read the introduction up to this point and your are still uninterested, then ok! I don't worry.

If you are still in two minds about this thesis topic, then please read on. I

am going now to discuss some interesting points regarding *bigP* in special and primes in general.

10.1.4 Really big numbers

It seems that bigP is a really big number, isn't it?

Well, this depends. Clearly, compared to numbers we are usually dealing with in economic applications, *bigP* appears to be really big. After all, it has about 22 mill. decimal digits! So, let us have a look at physics, that science which connects the micro cosmos of quantum world to the macro cosmos of our universe. In 1938 Arthur Stanley Eddington argued that the total number of protons in the whole universe is about 10^{80}, plus/minus a few percent. Note that one percent of 10^{80} is 10^{78}! Still quite a big number, but nevertheless ridiculously small compared to *bigP*.

Coming back to number theory, in a very real sense, there are no big numbers: any explicit number can be said to be *small*. Indeed, no matter how many digits or towers of exponents you write down there are only finitely many numbers smaller than your candidate and infinitely many that are larger (Crandall et al., 2005, p. 2). This is a rather general reservation. But even if we leave it aside, *bigP* is not really a record holder regarding size.

Very impressive in size is the *Skews Number* $10^{10^{10^{34}}}$ which in the 1950s played some role in prime number theory. This number is so big that if we could manage to somehow materialize each of its decimal digits to a subatomic particle our whole universe would be too small to hold it. The strange numbers $googol = 10^{100}$ and $googolplex = 10^{googol}$ are both smaller than the Skews number - but:

$$googol < bigP < googolplex$$

10.1.5 Mersenne numbers

The number bigP is a Mersenne number

And this is no exception: among the largest known primes the first 12 are all Mersenne numbers. So, what are these special numbers?

For any number $n \geq 0$ the n-th Mersenne number is defined by

$$M_n = 2^n - 1, \qquad n = 0, 1, 2, \ldots$$

These numbers are named after the french monk Marin Mersenne who studied them in the 17th century. M_n can be prime only if n is prime. For, if $n = ab$ is composite, then $2^{ab} - 1$ is divisible my $2^a - 1$ and $2^b - 1$. It is easy to see that: just use the well known formula for the sum of a finite geometric progression. Let us have a look at a few Mersenne numbers with prime exponent:

n	2	3	5	7	11	13	17	19	23
$M_n = 2^n - 1$	3	7	31	127	2047	8191	131071	524287	8388607

This table shows us that primality of n does not guarantee primality of M_n. Indeed, in the second row of this table there are two composite numbers:

$$M_{11} = 2047 = 23 \cdot 90, \qquad M_{23} = 8388607 = 47 \cdot 178481,$$

as you can check easily with your pocket calculator.

But the fact that among the record holders only with a few exceptions all primes have been Mersenne numbers may give rise to the suspicion that Mersenne numbers have some strong *affinity* to being prime.

Is this really so? *No!* Indeed, it is not even known whether there are infinitely many Mersenne primes. But it is known that these special primes are rare among all primes. To give you an impression: the two today largest known primes are:

$$\begin{aligned} bigP = M_{74\,207\,281} &= 2^{74\,207\,281} - 1, \quad \text{found in 2016} \\ M_{57\,885\,161} &= 2^{57\,885\,161} - 1, \quad \text{found in 2013} \end{aligned}$$

We can estimate the fraction of primes that lie between $M_{57\,885\,161}$ and $M_{74\,207\,281}$. This fraction is about $1 - 10^{-5\,000\,000} \approx 1$. So practically all primes $\leq M_{74\,207\,281}$ lie in this interval. But officially[2] none of these enormous number of primes is known. I will show you in a moment how we can arrive at this estimate.

One reason why Mersenne primes figure so prominently in prime number search is that there are quite efficient methods to check primality of these numbers. This brings us to the next point.

10.1.6 Primality testing

How can we know that bigP is indeed a prime number?

Let us state this question in more general form:

Given a number n, how can be find out that n is prime?

In computational number theory this problem is known a *prime number testing* or *primality testing*. Many algorithms are known to solve this problem and, after all, today we know that this task is *tractable* in the sense that there are efficient methods for primality testing.

A very simple approach is *trial division*. It runs as follows:

- Scan a list of all primes $\leq \sqrt{n}$. Note, that if n is composite, i.e. $n = a \cdot b$, then one of its factors must be $\leq \sqrt{n}$.
- If a prime p_i in this list divides n, then we know for sure that n is composite and therefore no prime number.

Although easy to understand and easy to implement this algorithm can be used only as long as n is of moderate size, say $n < 10^6$. We can estimate the amount of computational work required in the worst case, i.e. when n proves to be

[2] NSA may know more than we do!

prime. To obtain this estimate we use a really marvelous result from number theory, the celebrated *Prime Number Theorem (PNT)*. Let $\pi(n)$ denote the number of primes $\leq n$, then for large n:

$$\pi(n) \sim \frac{n}{\ln n} \tag{PNT}$$

The $\sim$-sign tells us that the ratio of right hand side and $\pi(n)$ tends to 1 as $n \to \infty$. In other words, the relative error of this approximation goes to zero.

Lets give it a try: is 1000003 prime? We have $\sqrt{1000003} \approx 1000$ and PNT tells us that the worst case number of trial divisions is about 145 (exact value: 168). So, this is tractable.

But for *bigP*? We estimate the worst case number of trial divisions by PNT as

$$\pi\left(\sqrt{bigP}\right) \sim 10^{10^7}$$

No chance to test *bigP* by trial division!

Note that when we use trial division to test primality of some number n and it turns out that n is composite, then we get even more: we get also a prime factor of n!

Interestingly, we can test *compositeness* of n without knowing any of its prime factors. Basic to most approaches of this type is another celebrated result from number theory, *Fermat's Little Theorem*.

First, let us state this theorem in rather informal terms:

If p is prime, then for any number a with $1 < a < p$, $a^{p-1} - 1$ is divisible by p.

Let's give it a try and put $p = 11$ and $a = 3$. Then the theorem says that $3^{10} - 1$ must be divisible by 11. Indeed,

$$3^{10} = 59049, \quad \text{and} \quad 3^{10} - 1 = 59048 = 11 \cdot 5368$$

It is time now to introduce an extremely elegant and useful formalism invented by C. F. Gauss, *modular arithmetic*. At its heart there is the idea of *congruence*. Given three integers[3] a, b and m, we write

$$a \equiv b \pmod{m} \tag{10.1}$$

This has to be read as: a is *congruent to b modulo m* and that means: the difference $a - b$ is divisible my m. For instance

$$16 \equiv 2 \pmod{7},$$

because $16 - 2 = 14$ is divisible by 7. Similarly $42 \equiv -3 \pmod{9}$, etc. Congruences behave almost like equations, they can be added, subtracted and multiplied. Even, if $a \equiv b \pmod{m}$, then

$$a^n \equiv b^n \pmod{m}, \qquad \text{for } n > 0. \tag{10.2}$$

[3] so not necessarily positive

Congruences may also be divided, but division is special.

The mod in (10.1) should not be mixed with the arithmetic operator mod. This is a very handy tool, a *binary* operator, defined as:

$$a \bmod b = \text{remainder left when } a \text{ is divided by } b \tag{10.3}$$

For instance, $7 \bmod 5 = 2$, because $7 = 5 \cdot 1 + 2$. In some computing environments and programming languages, e.g., in C, there is the special symbol %, the percentage-sign, for the mod-operator.

Let us now restate Fermat's Little Theorem using a congruence:

Fermat's Little Theorem: If n is prime, then for any $1 < a < n$

$$a^{n-1} \equiv 1 \pmod{n} \tag{10.4}$$

(10.4) is also known as *Fermat Test*.

However, there is a problem when using (10.4) as a primality test. If n is composite, then the test will fail ((10.4) is not satisfied) and signal that n is not prime. For instance, let's try $n = 81$ with base $a = 2$. Then we find

$$2^{80} = (2^{16})^5 \equiv 7^5 = 16807 \equiv 40 \not\equiv 1 \pmod{81}.$$

So we can be safe that $n = 81$ is not prime. Note that applying the Fermat Test requires to compute very high powers modulo some big number n. You may wonder how to do this. But it's easy once you know how to work with congruences and the mod-operator, and once you know, how to compute high powers such like 2^{1008} using the method of *repeated squaring*.

Pierre de Fermat
(1607-1665)

What about the other way round? If a number n passes the *Fermat Test* with a given base a, can we be sure that n is prime?

Unfortunately, no! For instance, $n = 15$ is clearly a composite number, but, with $a = 11$ we find:

$$11^{14} = (11^2)^7 = 121^7.$$

Now by the rules of modular arithmetic

$$121 \equiv 1 \pmod{15},$$

so

$$121^7 \equiv 1^7 \equiv 1 \pmod{15}.$$

Thus with base $a = 11$ $n = 15$ passes the *Fermat Test*. Composite numbers passing (10.4) are called *Fermat pseudo primes* with respect to base a. Using

another base may yield another result. If we use $a = 2$ instead of 11 as base, we obtain

$$2^{14} \equiv 4 \pmod{15}, \quad \text{(please verify!)}$$

and $n = 15$ is correctly identified as composite. However, it may happen that even if we try all possible bases a, the number n may pass the *Fermat Test* for each choice of a and may still be composite. Indeed, there are infinitely many numbers having this property, the *Carmichael numbers*. The smallest Carmichael number is $561 = 3 \cdot 11 \cdot 17$ and it will pass the *Fermat Test* with each base $1 < a < 560$. Thus, *Fermat's Test* is not really conclusive. If a number n passes the test, all we can say: *n is probable prime*.

That's an idea! Randomness may help. Indeed, there is a simple work-around to remedy the fuzziness of *Fermat's Test* which can be crafted into a *probabilistic* algorithm to test primality. This is the famous *Miller-Rabin Test*. It is a *Monte Carlo Algorithm*, thus an probabilistic algorithm yielding in reasonable time an answer which is correct with high probability[4]. There do exist *deterministic* algorithms, e.g. the AKS-test, but interestingly, these procedures are not competitive at the current state of the art. However, for our *bigP* the situation is different as it is a very special prime, a *Mersenne number*, and for such special numbers there is a very efficient test, the *Lucas-Lehmer Test* which is surprisingly easy to carry out.

10.1.7 Generating prime numbers

Let's return to the security server of *Amazon*, say. When initiating a new session with a registered customer the server requires two big primes p and q to calculate $n = p \cdot q$, the public key. Typically these primes are of order 2^{2048} (more than 600 decimal digits). But where do p and q come from? One possibility is to generate a reasonably large list of secret primes and store them in a file so that p and q are readily available on request.

This is not a good idea: (a) firstly, it is potentially unsafe. Image the data file being hacked? Secondly, the *Prime Number Theorem* tells us that there are simply too many primes to be stored.

Thus in practice the primes required by RSA are generated on-line. But how is this done?

To approach this issue let us for the moment bake smaller buns. Actually there is an algorithm which is already in use for about 2500 years, the *sieve of Eratosthenes*. Given a number n it creates a list of all primes $p \leq n$. Although easy to apply, it is not very efficient and in practice it is only used up to 10^6, sometimes even up to $n = 10^{12}$. These numbers are certainly too small to guarantee security of RSA. The solution is to adapt the *Miller-Rabin Test* mentioned above to generate *industrial grade primes*, strictly speaking numbers which are prime with sufficiently high probability.

[4] In contrast, a *Las Vegas algorithm* is also probabilistic and always yields the correct answer, but it may run for a very long time to find this answer.

10.1.8 Factoring of integers

The *Fundamental Theorem of Arithmetic* says that every number > 1 has a factorization into prime factors which is unique up to the ordering of factors. E.g.,

$$n = 12345654321 = 3^2 \cdot 7^2 \cdot 11^2 \cdot 13^2 \cdot 37^2. \tag{F}$$

You may notice that this *palindromic number* n turns out to be a perfect square!

But how do we arrive at a representation (F)? Now the story is becoming really thrilling!

As long as the composite n is not too big, say $< 10^6$, maybe $n < 10^{12}$, a simple adaptation of the sieve of Eratosthenes does the job. But for large composites, composites of *industrial size*?

The state of the art is this: factoring a composite number is *believed* to be very a hard problem. This is, of course not true for all composites – those having small factors are easy to factor. But in general, the problem seems to be very difficult. Remarkably though, the only evidence that this statement is true is our apparent inability to find fast algorithms[5]. So it is quite surprising that an entire industry is based on the belief that factoring of integers is hard, indeed. The current record is a number known as *RSA-768*, a number with 232 decimal digits. It has been factored in 2009 using a CPU time of about 2 years!

In your thesis you should not discuss factorization methods for numbers that large, of course. It is sufficient to consider numbers n such that $2^{32} < n < 2^{64}$.

Once again we come across the name *Fermat*. He has devised a general purpose algorithm, known as *Fermat's Method*, which is applicable when n has two prime factors being not too far apart. I.e., $n = p_1 \cdot p_2$, $p_1 < p_2$, both odd numbers and $p_2 - p_1 = 2d$, where d is a relatively small number. In this case we can write

$$n = (x - y)(x + y) = x^2 - y^2, \quad \text{where } x = p_1 + d, \quad y = d.$$

The proper x can be found by successively trying

$$x = \lfloor \sqrt{n} \rfloor,\ \lfloor \sqrt{n} + 1 \rfloor, \lfloor \sqrt{n} + 2 \rfloor, \ldots \tag{Q}$$

Here $\lfloor x \rfloor$ denotes the *floor-function* which rounds down x the the next smallest integer.

We continue (Q) until we find that $x^2 - n$ is a perfect square. In that case $y^2 = x^2 - n$.

Here is an example: let $n = 8537529$. Then

$$x = \lfloor \sqrt{8537529} \rfloor + 1 = 2922,$$

[5] Well, there is Shor's Algorithm, a Monte Carlo methods not unsimilar to the Miller-Rabin Test. But Shor's Algorithm needs a quantum computer to run on. May be your are aware of the comedy series *The Big Bang Theory*? In season 1, episode 13, *The Bat Jar Conjecture*, there is a competition, the Physics Bowl. One of the questions posed by Dr. Gablehauser was about Shor's Algorithm. Incidentally, Leslie Winkle knew the answer.

and successively:

$$x = 2922 \qquad x^2 - n = 555$$
$$x = 2923 \qquad x^2 - n = 6400 = 80^2$$

Here we stop as we have found a perfect square $y^2 = 80^2$. Thus a nontrivial factorization of n is found to be:

$$8537529 = (x - y)(x + y) = (2923 - 80)(2923 + 80) = 2843 \cdot 3003$$

In this example *Fermat's Method* worked very fine. But this is not always the case. Indeed, we can find easily examples where *Fermat's Method* is outperformed drastically be simple trial division. However, *Fermat's Method* may be taken as starting point to develop considerably more efficient algorithms, among these the *Quadratic Sieve* due to Carl Pomerance. By the way, like in prime testing algorithms *randomness* is an option and comes to our help. A nice example is *Pollard's* $\rho - 1$ *Method* is a Monte Carlo Algorithm being quite effective, if the factors of n are not too large. There is also a $p - 1$ *Method* due to Pollard which makes again use of *Fermat's Little Theorem*.

So far a short introduction to the topic.

If you've read up to this point and you are still interested in this topic, then welcome on board!

10.2 Where to go from here

Write an interesting thesis in nice style so that people not specialized on this topic keep on reading simply because you could raise their interest in this area. Also, your thesis should be a fine mixture of theory and application. It wouldn't be a good idea to separate the material you want to present in two major blocks, one devoted to elementary theory, and one devoted to computational problems.

Of course, a major portion of your work should be devoted to computational number theory. This means that you should *implement* various algorithms and present examples showing how they work.

10.2.1 Computational issues

I am certainly aware of the fact that this point is by no means a trivial one. General purpose computing environments like R, Matlab or its free clone octave are not suitable, because integers when they undergo changes resulting from diverse calculations are usually casted into floating point numbers. So they are no longer integers internally. This in turn means that you inevitably run into serious problems with round off errors which hardly can be controlled. Therefore it will be necessary to use a computing environment which knows about the *integer data type*, e.g. C, C++, Phyton have this. Thus it is possible

to handle numbers up to $2^{64} - 1 = 9223372036854775807$. I will be perfectly satisfied when you can manage this.

Indeed, there is also GMP, Gnu's Multiprecision Library which can be linked to C, C++, Python and Java. Using GMP you can work with integers of practically arbitrary size, the only limit is imposed by the availability of main memory. It is also possible to call GMP from within R.

There is a second route you can pursue for numerical calculations. You may use directly a computer algebra system like Mathematica, Maple or SAGE. The first two in this list are commercial products, however, student's licenses are available. For Mathematica there is a WU campus license, as far as I know.

SAGE is special in two respects: it is a freely available CAS-project with a strong flavor of computational number theory, and William Stein, initiator of SAGE is working intensively in number theory. Here you can learn more about SAGE:

`http://www.sagemath.org/`

The CAS just mentioned work with infinite precision numbers as a built-in data type. And of course, they supply lots of functions on computational number theory. BUT, and this is considered a BIG BUT: when using Mathmetica, Maple or SAGE, you must implement your algorithms using the programming facilities offered by these CAS. This is for pedagogical reasons, as I want you to *learn how things really work*.

10.2.2 Issues of general interest

These points should be take care of in your thesis, theoretically and computationally:

- Euclid's Theorem that there are infinitely many primes.
- The Fundamental Theorem of Artithmetic, i.e., each number has a unique prime factorization.
- The greatest common divisor of two numbers. Finding the greatest common divisor of two numbers can be performed very efficiently by one of the oldest algorithms still in use today. Euklid's Algorithm is most often used in an *extended* form. You should take care of this and mention also *Bézout's Lemma*, as this turns out to be very important in number theory.
- Give a careful exposition to *modular arithmetic*. You will find out that this is not only easy to grasp but also extremely useful, if not to say indispensable.
- State and prove *Fermat's Little Theorem*. You will find that there are many proofs, even combinatorial ones.
- Discuss and implement the algorithm for *repeated squaring* which is the most efficient way to calculate high powers of integers modulo some given number (mostly a prime).

- You should also discuss how to determine whether a number is a perfect square by an adaptation of Newton's Method.

10.2.3 Some more suggestions

Here you can make your personal choice on what topics you will put your emphasis. This may be prime testing, generation of primes or even the factorization problem. Of course you may also discuss RSA. But, you should not make cryptology to the major topic of your thesis, as this is a very different story and would lead you to far afield.

10.2.4 What to be avoided

These are some points which you should not cover in your theses unless you know what you are doing. The reason for avoiding these points is simply, the theory behind them is much too difficult.

- Do not go into too much detail about *quadratic residues*.
- Avoid methods based on *elliptic curves*. These methods belong to the most powerful in computational number theory but also to the most complicated.
- You may state and use the *Prime Number Theorem*, but do not try to give a proof. Standard proofs of this fundamental result make heavy use of complex function theory, although there are also proofs not relying on complex analysis, but these are very messy.

10.3 An Annotated Bibliography

Nice and very readable introductions to number theory are the books of Andrews (1994) (chapters 1, 2, 4 and 8) and Hardy and Wright (2008). The latter book is the classical text book on number theory. The mandatory material mentioned above is covered in chapters 1, 2, 5 and 6. In these texts you will also find smooth introductions to modular arithmetic. Another introductory text on modular arithmetic is a paper by Kak (2016).

The most important text book for your thesis is Crandall et al. (2005). This text gives very good introduction to computational aspects of prime numbers. You will find there algorithms for prime testing, prime number generation and factoring. Indeed, most of the introduction to this thesis has been inspired by this book. There is also a free download version available.

The booklet by Rempe-Gillen and Waldecker (2014) contains a lot of interesting material on basic concepts of number theory and on primality testing. In particular you will find there also a discussion of the *AKS-Algorithm*.

From its appearance Wagstaff (2013) looks like a book with fairy tales for children. But beware! It is a very serious and up-to-date textbook on the

integer factoring problem. However, the author presupposes from his readers some basic knowledge on basic number theory, so the text is not so easy to read and to comprehend. Having read some introductory texts mentioned above will be helpful. Still it is an extremely valuable text. Particularly interesting are chapters 2, 4 and 5. May be you find also chapter 10 worth reading as it discusses some *dirty tricks* in factoring.

A very nice survey paper on the factoring problem is Pomerance (1996). The author is inventor of the *Quadratic Sieve*, one of the most powerful algorithms for integer factoring. Also, if you want to put emphasis on integer factoring, then strongly recommended is the paper by Lenstra (2000).

Finally, some interesting texts on number theory and cryptography. The classical paper on public key encryption and related problems like digital signatures is, of course, Diffie and Hellman (2006). A fine text book is Buchmann (2001), interesting material on this topic is presented also in Wagstaff (2013). And I should also mention *The Codebreakers* by David Kahn (1996). This great book is a comprehensive history of cryptology, exciting like a detective novel.

10.4 References

[1] G. E. Andrews. *Number Theory*. Dover Books on Mathematics. Dover Publications, 1994.

[2] Johannes A. Buchmann. *Introduction to cryptography*. Undergraduate texts in mathematics. Springer, 2001.

[3] Richard Crandall et al. *Prime numbers: a computational perspective. Second Edition*. 2nd. Springer, 2005.

[4] W. Diffie and M. Hellman. "New Directions in Cryptography". In: *IEEE Trans. Inf. Theor.* 22.6 (Sept. 2006), pp. 644–654.

[5] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. 6th. Oxford University Press, 2008.

[6] Avi Kak. *Modular Arithmetic*. 2016. URL: https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture5.pdf.

[7] Arjen K. Lenstra. "Integer Factoring". In: *Des. Codes Cryptography* 19.2/3 (2000), pp. 101–128.

[8] Carl Pomerance. "A tale of two sieves". In: *Notices of the American Mathematical Society* 43 (1996), pp. 1473–1485.

[9] Lasse Rempe-Gillen and Rebecca Waldecker. *Primality Testing for Beginners*. American Mathematical Society, 2014.

[10] Samuel S. Wagstaff. *The Joy of Factoring*. American Mathematical Society, 2013.

Topic 11

Elementary Methods of Cryptology

No matter how resistent the cryptogram, all that is really needed is an entry, the identification of one word, or of three or four letters.

Helen Fouché Gains, 1939

Keywords: cryptography, cryptanalysis, substitution ciphers, transposition ciphers, Monte Carlo Markov Chains, simulated annealing

11.1 An Invitation

One of may favorite books is *Mathematical Recreations and Essays* by Rouse Ball and Coxeter (1987). Mathematical recreations? Seems to be a contradiction in terms. But believe me, this is not so. The very last chapter of this book deals with cryptology and it begins with these remarkable sentences:

> *The art of writing secret messages – intelligible to those who are in possession of the key and unintelligible to all others - has been studied for centuries. The usefulness of such messages, especially in the time of war, is obvious; on the other hand, their solution may be a matter of great importance to those from whom the key is concealed. But the romance connected with the subject, the not uncommon desire to discover the secret, and the implied challenge to the ingenuity of all from whom it is hidden have attracted to the subject the attention of many to whom its utility is a matter of indifference*[1].

Cryptology, the art and science of secret writing should be the topic of your thesis.

[1] These words are apparently due to Abraham Sinkov (1907-1998), an american mathematician with important contributions to cryptology.

11.1.1 Some basic terms

Let us start fixing some important terms. Cryptography is that field of cryptology which deals the understanding and implementation of techniques to obfuscate information. These techniques are usually called *cryptographic algorithms*, *cryptographic systems*, in short *cryptosystems* or *ciphers*.

The text whose meaning should be concealed is called the *plaintext*. When the rules of a cipher are applied to the plaintext, one says also, the plaintext is *encrypted*, the result is called the *ciphertext*. *Decryption* is the reverse process of recovering the plaintext from the known ciphertext. In many cases ciphers have to rely on an external piece of information, the *key*.

Cryptanalysis on the other side, is the art of *breaking* a cipher. Given a piece of ciphertext we want to recover the underlying plaintext usually without additional information. This is also called a *ciphertext only attack*.

Steganography is a related field. It provides methods with the definite aim to hide the existence of a secrete message at all. Examples are invisible inks, micro dots, changes of the color values of pixels in a digital image, etc.

In this thesis you should consider only ciphers which work an a *letter-by-letter* basis using a particular *plaintext alphabet*. For our purposes we shall consider only the 27-letter alphabet in the standard lexicographic ordering:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
␣	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Table 11.1: Our standard alphabet

where ␣ denotes the space-character. Let us call this our *standard alphabet*.

At first the plaintext to be enciphered has to be prepared: all letters are turned to lowercase, all punctuation is removed, special characters like *diereses* are expanded, e.g. *ä* becomes *ae*, numbers are translated into appropriate numerals, e.g. *12* becomes *twelve*, etc. Furthermore, multiple spaces are condensed into single spaces and specials characters like newlines and tab-stops are removed. Note that we retain ␣, the space character. In general this is not a good idea as this may become a severe weakness of a cipher, but we shall keep spaces because of readability, and also because it makes more fun.

The ciphertext resulting from applying a particular cipher and a key to a given plaintext should be assumed to use the same alphabet as plaintext but all letters written in uppercase. Examples follow.

According to the way plaintexts are transformed to ciphertext general cryptographic systems are divided into two classes.

- *Substitution ciphers*. In these systems letters change their values. For instance, an `a` at some position in plaintext may be changed to `W`, at another position `a` may be changed to `M`, etc.

Example 1.

plaintext	`send me more money`
ciphertext	`KABVDSNSUGNTRQUWXF`

There are two points to observe in this example: (1) look at the letter `e` in plaintext. The first `e` is mapped to a `A`, the second occurrence of `e` maps to `N`, etc. (2) ciphertext and plaintext have the *same length*, however, this need not be so.

Monoalphabetic substitution ciphers use one alphabet only, this means, that a particular plaintext letter is always mapped to the same ciphertext letter. *Polyalphabetic* substitution ciphers use several alphabets and switch between them according to some rule. Example 1, for instance, is polyalphabetic.

- *Transposition ciphers.* Letters retain their value but change position.

  Example 2.

plaintext	`send me more money`
ciphertext	`YENOM EROM EM DNES`

  Whereas this cipher is easily decrypted (just be inspection) the cipher in Example 1 is much more difficult.

In practice, many modern cryptographic systems make heavy use of both substitution and transposition. A typical example is the AES system. However, in this thesis we should stay at rather elementary systems. Some of them will be now presented.

Lets start with substitution ciphers.

11.1.2 Caesar's Cipher

Encryption and decryption

This is the simplest case of a monoalphabetic substitution cipher. The roman historian Suetonius reports that Caius Iulius Caesar used this extraordinarily simple system to encrypt messages about battle orders, movements of military forces, etc.

Encryption is performed using a *translation table* in which the ciphertext alphabet results from the plaintext alphabet by a simple circular shift of d positions to the left. This number d is the *key* of the system. Caesar mostly used the key $d = 3$. The corresponding translation table is then:

_	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B

Table 11.2: The translation table of a Caesar's cipher

Example 3. Here's some plaintext message and the corresponding ciphertext with unknown key d:

```
come home again all is forgiven
MYWOJRYWOJKQKSXJKVVJSBJPYAQSEOX
```

It's interesting to observe that Caesar's cipher is even used in our days! You may have seen the science fiction movie *2001: A Space Odyssey*. The name of the rogue computer *HAL* is just a Caesar-cryptogram of *IBM*!

Decryption of a Caesar's cipher is very easy. Given the key d perform a circular shift by d letters to the *right*.

As we will need this later, we show now that encryption and decryption can be expressed very conveniently in *algebraic form*. For this purpose we use the *mod* binary operator:

$$a \bmod b = \text{remainder of dividing } a \text{ by } b$$

For instance:

$$12 \bmod 7 = 5, \quad 4 \bmod 13 = 4, \quad 2^{7000} \bmod 7001 = 1, \quad \text{etc.}$$

Some care is needed when dealing with negative numbers. The correct definition of the *mod* operator is found in Graham, Knuth, and Patashnik (2003, p. 82):

$$a \bmod b = a - b\lfloor a/b \rfloor, \tag{11.1}$$

where the floor function $\lfloor x \rfloor$ gives x rounded *down* to the next smallest integer. Thus

$$(-3) \bmod 27 = -3 - 27\lfloor (-3)/27 \rfloor = -3 - 27 \cdot (-1) = 24 \tag{11.2}$$

Note in passing. Programming languages like C, Python or Java have a mod-operator denoted by `%`. Only Python's version behaves for negative values correctly like (11.1). C and Java return -3 in (11.2). When implementing ciphers in C or Java you should take care of this feature.

To encrypt a message using Caesar's Cipher with key d we do simply the following:

- By Table 1, map each plaintext letter a_i to an integer $0 \leq a_i \leq 26$.
- Calculate the ciphertext letter c_i corresponding to a_i by

$$c_i = (a_i + d) \bmod 27$$

  Translate the numbers c_i back to letters by Table 1 to get the ciphertext.

Example 3 (continued) With key $d = 10$:

Plaintext	c	o	m	e	_	h	o	m	e	_	a	g	a	i	n	_	a	l	l	_	...
	3	15	13	5	0	8	15	13	5	0	1	7	1	9	14	0	1	12	12	0	...
	13	25	23	15	10	18	25	23	15	10	11	17	11	19	24	10	11	22	22	10	...
Ciphertext	M	Y	W	O	J	R	Y	W	O	J	K	Q	K	S	X	J	K	V	V	J	

Decryption is the inverse, just *subtract* the key d and take care of negative numbers using (1):

$$a_i = (c_i - d) \bmod 27$$

Cryptanalysis of Caesar's cipher

Is Caesar's cipher a secure one? It is customary to assess this important point by looking at the *key space* $\mathcal{K}$. This is the set of all possible keys the system accepts. The number of keys $|\mathcal{K}|$ is a measure of computational work which is necessary in the following *worst case scenario*: if we have only the ciphertext and want to break the system by *brute force*, then in the worst case $|\mathcal{K}| = 26$ keys have to tested. This is a very small number, so Caesar's system cannot be considered secure.

Brute force is actually the method of choice for this cipher, we just start width $d = 1, 2, \ldots$ and try all keys, rotating each time the standard alphabet to the right by d places.

But two important issues come immediately into our mind:

- How can we know that a particular ciphertext was created by a particular cryptographic system, in our cases Caesar's system?
- Even if we know, how can we find out the language of plaintext so that we are able so recognize the unknown key?

There do exist important cryptanalytic tools that allow us to find reasonable answers to these questions, provided the available ciphertext is not too short. See Section 2 for more about that.

Let's continue Example 3 and perform a brute force attack by systematically trying keys $d = 1, 2, \ldots$ to decrypt the ciphertext. We obtain:

$d = 1$:	LXVNIQXVNIJPJRWIJUUIRAIOX PRDNW
$d = 2$:	KWUMHPWUMHIOIQVHITTHQ HNWZOQCMV
$d = 3$:	JVTLGOVTLGHNHPUGHSSGPZGMVYNPBLU
	...
$d = 10$	COME HOME AGAIN ALL IS FORGIVEN

Of course, this wasn't a challenge. But still we may ask: *Is there a way to find the key d without brute force?*

11.1.3 Frequency analysis

This is one of the most important concepts in cryptanalysis and you should take care of it in your thesis by a thorough discussion.

Human languages, may it be English, German, but also artificial languages like Esperanto or even Klingon, follow certain rules. Words are not merely random combinations of letters. Indeed, human languages exhibit certain *statistical regularities* which can be identified by appropriate methods. The simplest and oldest method is to calculate the *frequency distribution* of letters in a text. This idea is to Al Kindi (801 - 873 AD) an Arab mathematician and philosopher.

To apply frequency analysis we first need a *learning sample* to obtain the reference distribution of letters. Usually we look for a sufficiently large *text corpus* and count the occurrences of various letters. Of course, this requires also that we have some idea about the language of the unknown plaintext.

For the purpose of this introduction I selected Tolstoj's *War and Peace*. The text (more than 3 million letters) has been prepared in the sense described above. Then I counted letters in *War and Peace* and in the cryptogram of Example 3. The results are shown in Table 3 below. It may be more informative to make a line plot of these frequency distributions (See Figure 1).

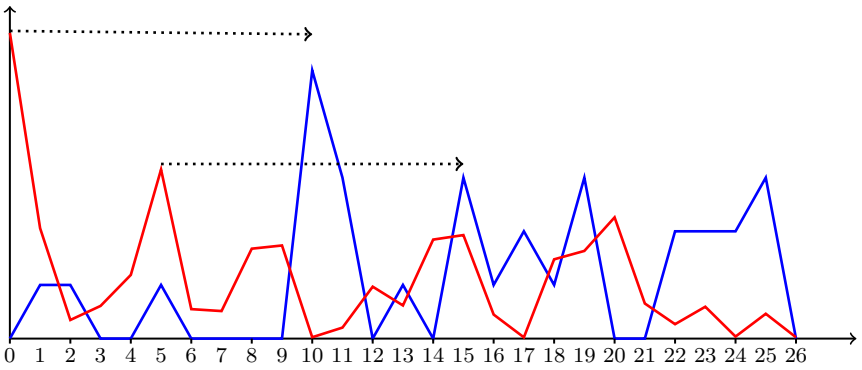


Figure 11.1: The frequency distributions of Table 3

Here we can see very clearly:

- The letter with highest frequency in *War and Peace* is the space character with about 18 %. In the cryptogram the letter with highest frequency is `J`. This strongly suggests a shift of $d = 10$ between these distribution. Thus we may try as a first guess the key $d = 10$.
- The same pattern can be seen if we compare the letters with second highest frequency: `e` in the text corpus, `O` in the cryptogram, again a distance of 10.
- Also quite remarkable: the cryptogram has only 31 letters, still it seems that the frequency comparisons are conclusive.

Observe that our analysis above is based on a simple visual inspection of the frequency plots, just look at the peaks!

	War And Peace			*ciphertext*	
n	character	abs. freq.	rel. freq.	abs. freq.	rel. freq.
0	space	565 454	0.1834	0	0.0000
1	a	204 128	0.0664	1	0.0323
2	b	34 419	0.0112	1	0.0323
3	c	60 448	0.0197	0	0.0000
4	d	117 752	0.0383	0	0.0000
5	e	312 716	0.1017	1	0.0323
6	f	54 491	0.0177	0	0.0000
7	g	50 906	0.0166	0	0.0000
8	h	166 293	0.0541	0	0.0000
9	i	172 223	0.0560	0	0.0000
10	j	2 485	0.0008	5	0.1613
11	k	20 322	0.0066	3	0.0968
12	l	96 030	0.0312	0	0.0000
13	m	61 286	0.0199	1	0.0323
14	n	183 114	0.0595	0	0.0000
15	o	191 440	0.0622	3	0.0968
16	p	44 456	0.0145	1	0.0323
17	q	2 319	0.0008	2	0.0645
18	r	146 594	0.0477	1	0.0323
19	s	162 126	0.0527	3	0.0968
20	t	224 202	0.0729	0	0.0000
21	u	64 911	0.0211	0	0.0000
22	v	26 641	0.0087	2	0.0645
23	w	58 925	0.0192	2	0.0645
24	x	3 758	0.0012	2	0.0645
25	y	45 931	0.0149	3	0.0968
26	z	2 387	0.0008	0	0.0000
Total		3 075 757	1.0000	31	1.0000

Table 11.3: Frequency Counts

Is there a better way to compare two discrete distributions?

There are quite a number of ways we can do better. A very simple idea is to calculate the *total variation distance* between two distributions.

Consider two frequency distributions f_i and g_i defined on the same set $\mathcal{X}$. In our case

$$\mathcal{X} = \{0, 1, 2, \ldots, 26\}$$

We say that f_i and g_i are *close* if the total variation distance is small. The latter is defined by

$$\|f - g\|_{TV} = \frac{1}{2} \sum_{i \in \mathcal{X}} |f_i - g_i|$$

I've calculated $\|f - g\|_{TV}$ and plotted for all shifts $d = 0, 1, \ldots, 26$. You can see the striking down-peak at $d = 10$ in Figure 2. Again we measure a strong signal indicating that the key is $d = 10$. Note that this can be found out by the computer in a more or less *automatic fashion* and does not need human intervention.

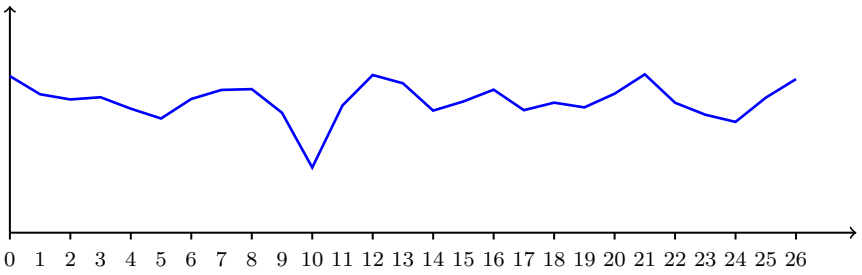


Figure 11.2: TV of distributions in Table 2 for shifts $d = 0, 1, \ldots, 26$

11.1.4 Monoalphabetic substitution

Encryption and decryption

Caesar's Cipher is a special case of a monoalphabetic substitution cipher. The latter is defined by a translation table for the standard alphabet (Table 1) where in the second row we have a *permutation* P of first row. In case of Caesar's Cipher this permutation is *cyclic*, as it is obtained by a cyclic shift of letters. But now we no longer require this permutation to be cyclic. As a result the classical monoalphabetic substitution is much stronger than Caesar's cipher.

Example 4. Given is the following translation table where the second row is a random permutation of the standard alphabet.

plaintext alphabet:	_	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext alphabet = key:	Y	C	S	O	E	X	P	Q	Z	B	W	L	H	V	M	D	G	U	J	K	A	T	I	_	F	R	N

Table 11.4: A translation table for a monoalphabetic substitution cipher

Enciphering is easy. For example assume that we want to conceal the somewhat desperate message *need reinforcements at once*:

plaintext:	`need reinforcements at once`
ciphertext:	`MXXEYJXBMPDJOXVXMAKYCAYDMOX`

Observe that the *encryption key* is just the second row of the translation table.

Since our standard alphabet has 27 letters, the key space $\mathcal{K}$ consists of all permutations of 27 elements which is quite a lot:

$$|\mathcal{K}| = 27! = 10888869450418352160768000000 \approx 10^{28}$$

Applying a *brute force* attack would require to test so many keys in the *worst case*. But even if we have the best high-speed computers at our disposal, it will simply take too much time to break such a cipher by *brute force*.

Thus, if we accept the size of the key space as an indicator of secureness of a cryptographic system, then monoalphabetic substitution seems to be pretty safe. Later we will find out that this system is a rather weak one and cryptanalysis poses no real challenge for an experienced cryptanalyst.

But before we turn to these aspects: How can be decrypt a secret message when we possess the key?

This is easy. Just form the *inverse permutation* P^{-1} for the key P: sort the translation table along the second row and then interchange row 1 and row 2:

_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
w	t	i	a	o	d	x	p	l	v	r	c	k	u	z	c	f	g	y	b	u	q	m	j	e	_	h

Table 11.5: The translation table for deciphering Example 4

Using this table decryption is straightforward.

Cryptanalysis of a monoalphabetic substitution cipher

Due to the size of the key space a brute force attack using the ciphertext only is certainly not feasible.

What about *frequency analysis*?

Why not? A monoalphabetic substitution always maps a plaintext letter to the same ciphertext letter. Thus letter frequencies are preserved, but they are distributed *very* differently according to the key permutation P. Thus if e.g., the space character _ is mapped to the ciphertext letter W, then in a sufficiently long ciphertext W will be the letter with roughly the same frequency as the space character _ in plaintext. Normally, cryptanalysts just start this way and try to get a *clue* about the unknown plaintext by replacing the ciphertext letters with highest frequency by those letters which in a learning sample (e.g., *War and Peace*) have highest frequency.

Example 5. For the next example I have chosen a somewhat longer plaintext, let us call it T, with 714 letters. Upon enciphering I obtained the ciphertext[2]:

```
FNDEYQUUDXTDTHDGTDGYNDNHBDFNDEYQUUDOJXYGDJHDOLQHINDFNDEYQUUDOJXYGDTHDGYNDEN
QEDQHBDTINQHEDFNDEYQUUDOJXYGDFJGYDXLTFJHXDITHOJBNHINDQHBDXLTFJHXDEGLNHXGYDJ
HDGYNDQJLDFNDEYQUUDBNONHBDTSLDJEUQHBDFYQGNMNLDGYNDITEGDZQVDWNDFNDEYQUUDOJXY
GDTHDGYNDWNQIYNEDFNDEYQUUDOJXYGDTHDGYNDUQHBJHXDXLTSHBEDFNDEYQUUDOJXYGDJHDGY
NDOJNUBEDQHBDJHDGYNDEGLNNGEDFNDEYQUUDOJXYGDJHDGYNDYJUUEDFNDEYQUUDHNMNLDESLL
NHBNLDQHBDNMNHDJODFYJIYDJDBTDHTGDOTLDQDZTZNHGDWNUJNMNDGYJEDJEUQHBDTLDQDUQLX
NDRQLGDTODJGDFNLNDESWCSXQGNBDQHBDEGQLMJHXDGYNHDTSLDNZRJLNDWNVTHBDGYNDENQEDQ
LZNBDQHBDXSQLBNBDWVDGYNDWLJGJEYDOUNNGDFTSUBDIQLLVDTHDGYNDEGLSXXUNDSHGJUDJHD
XTBEDXTTBDGJZNDGYNDHNFDFTLUBDFJGYDQUUDJGEDRTFNLDQHBDZJXYGDEGNREDOTLGYDGTDGY
NDLNEISNDQHBDGYNDUJWNLQGJTHDTODGYNDTUB
```

Let us compare the three highest letter frequencies in the ciphertext with those in the text corpus (see Table 3):

War and Peace	_	e	t	*Ciphertext*	D	N	G
Frequency	0.1834	0.1017	0.0729		0.1966	0.1039	0.0716

[2] I shall keep the key secret!

So, it makes sense to try the substitutions

$$\texttt{D} \to \texttt{\_}\,, \quad \texttt{N} \to \texttt{e}\,, \quad \texttt{G} \to \texttt{t}$$

The result is interesting (to save space I have displayed only the first few lines):

```
Fe EYQUU XT TH tT tYe eHB Fe EYQUU OJXYt JH OLQHIe Fe EYQUU OJXYt TH tYe EeQE QHB
TIeQHE Fe EYQUU OJXYt FJtY XLTFJHX ITHOJBeHIe QHB XLTFJHX EtLeHXtY JH tYe QJL Fe
EYQUU BeOeHB TSL JEUQHB FYQteMeL tYe ITEt ZQV
```

- Suddenly we can see probable *word boundaries*!
- Other conjectures come into our mind: the word `tYe` is likely to mean plaintext `the`, which is the most common three-letter word in English. So we may try $\texttt{Y} \to \texttt{h}$.

```
Fe EhQUU XT TH tT the eHB Fe EhQUU OJXht JH OLQHIe Fe EhQUU OJXht TH the EeQE QHB
TIeQHE Fe EhQUU OJXht FJth XLTFJHX ITHOJBeHIe QHB XLTFJHX EtLeHXth JH the QJL Fe
EhQUU BeOeHB TSL JEUQHB FhQteMeL the ITEt ZQV
```

What about the word `EhQUU` which appears three times in the first line? May be, it means `shall`? Give it a try:

$$\texttt{E} \to \texttt{s}\,, \quad \texttt{Q} \to \texttt{a}\,, \quad \texttt{U} \to \texttt{l}\,:$$

```
Fe shall XT TH tT the eHB Fe shall OJXht JH OLaHIe Fe shall OJXht TH the seas aHB
TIeaHs Fe shall OJXht FJth XLTFJHX ITHOJBeHIe aHB XLTFJHX stLeHXth JH the aJL Fe
shall BeOeHB TSL JslaHB FhateMeL the ITst ZaV
```

Hm, looks interesting ...

However, from now on the work of the cryptanalyst becomes truly hard and messy. She has to try several *conjectures* about text snippets so that the text becomes closer and closer to English text, that the text becomes more *plausible*.

What cryptanalysts often do at this point is to form *contact tables*, meaning they gather statistics about the occurrence of *bigrams* in the ciphertext and compare these with statistics collected from a text corpus.

A *bigram* is just a 2-letter sequence in text. E.g., our original ciphertext begins with bigrams

`FN–ND–DE–EY` ...

What is lurking behind is a remarkable theory about human language. Early as 1906 A. Markov performed frequency counts of bigrams in Alexander Pushkin's *Eugene Onegin*. He used these to demonstrate and later prove an important extension of the *Law of Large Numbers* to *dependent trials*. That was the origin of one of the most important classes of *stochastic processes*, *Markov Chains*. This idea has been continued and extended by Claude Shannon in his foundation of a mathematical theory of communication (see the annotated bibliography at the end).

In *War and Peace* the most frequently occurring bigrams are:

Bigram	Frequency	Count
e_	0.0361	111116
_t	0.0285	87587
d_	0.0247	75995
he	0.0244	75022
th	0.0239	73400
_a	0.0226	69556
s_	0.0204	62865
t_	0.0189	58198
_h	0.0162	49954
in	0.0157	48180

You can see from these statistics that the letter `e` is most likely to occur at the end of a word, whereas `t` very often appears at the beginning of a word.

However, a by intuition guided process of trial and error as we applied when consulting frequencies of simple letters is very hard to carry out.

Is it possible to run the cryptanalytic process somehow automatically so that permanent human interventions can be avoided?

Yes! Here is a solution.

11.1.5 Combinatorial Optimization

Combinatorial optimization is a class of methods that deal typically with problems of very high dimension (= number of variables). In most cases of interest the variables are discrete and the space of possible solutions is finite-dimensional. Hence, in principle, it is possible to find the optimal solution by brute force, i. e., by *complete enumeration* of all solutions. But only in a few cases this strategy is viable, the number of admissible solutions is usually exorbitantly large, too large for such an unsophisticated approach. Indeed, most combinatorial optimization problems are *very hard* in a well defined mathematical sense.

The classic in combinatorial optimization is the *Traveling Salesman Problem (TSP)*: a salesman has to visit customers in n different cities. If the distances between each pair of cities is known, we have to find a tour such that

- each city is visited exactly once;
- the salesman returns to the city where his tour started;
- the tour has minimum length.

Technically, the problem reduces to finding a *permutation* of the numbers $1, 2, \ldots, n$ such that an *objective function* is minimized. Here the objective function assigns each permutation the total length of the corresponding tour.

Can you see parallels to the problem of breaking a monoalphabetic substitution cipher? It's the unknown key which is also a permutation of the standard alphabet! So, may be we can learn something from combinatorial optimization?

However, to find the optimum of a combinatorial optimization problem such as

the TSP one has usually resort to *iterative search procedures*. For this purpose several heuristic and meta heuristic algorithms have been developed. Among them one of the most successful is *simulated annealing*. A special adaptation of this meta heuristic is the *Metropolis Algorithm* which is very well suited for some cryptanalytic purposes.

First we need an *objective function*. In our case this will be a *plausibility measure* $f(p)$ defined on the set $\mathfrak{S}_{27}$ of all permutations (= keys) p of the standard alphabet. For this function $f(p)$ we require that it should preferably assume high values if the text deciphered with p is close to English text. We want to measure this by using bigram statistics in such a way that $f(p)$ takes on high values when the bigram frequencies in the decrypted text most closely match those of some reference text like *War and Peace*. Examples of plausibility measures may be found in Diaconis (2008) and Chen and Rosenthal (2010).

The Metropolis Algorithm runs roughly as follows:

- Fix a *scale parameter* $\alpha > 0$.
- Create an initial permutation p_0, e.g. a random permutation on $\mathfrak{S}_{27}$ and calculate the plausibility measure $f(p_0)$.
- Repeat the following steps for a sufficient number of iterations.
  - Given p_0 create a new permutation p_1 in a *uniform way* (to be explained shortly) and calculate the plausibility $f(p_1)$.
  - Sample a pseudo random number u having a uniform distribution on the interval $[0, 1]$.
  - if $u < \left(\frac{f(p_1)}{f(p_0)}\right)^{\alpha}$ then accept the new key p_1 : $p_0 \leftarrow p_1$. Otherwise reject p_1 and leave p_0 unchanged.

A few remarks are in order:

- If the new key p_1 yields higher plausibility than p_0, then $\left(\frac{f(p_1)}{f(p_0)}\right)^{\alpha} > 1$ and since $u \leq 1$, the better solution p_1 is always accepted.
- If the new key p_1 results in smaller plausibility than p_0, then p_1 may be still accepted with probability $\left(\frac{f(p_1)}{f(p_0)}\right)^{\alpha}$. This idea lies at the heart of simulated annealing: it allows us to escape a *local maximum* by temporarily accepting a worse solution.
- The scale parameter α, typically chosen close to 1, influences the probability of accepting a worse solution. It is closely related to the concept of *temperature* in simulated annealing.
- The new key p_1 can be created in many ways uniformly out of p_0. The most common technique is to select two different entries of p_0 at random and exchange them so to from the new key p_1. This is also called a *transposition*.

So, it's time to try that. I crafted in a more or less quick and dirty fashion an implementation of this algorithm in the C programming language and used

the plausibility measure proposed by Chen and Rosenthal (2010). Here are the results:

```
 200   HTEL ASSENUEUREYUEY TETRIEHTEL ASSEPON YEOREPMARVTEHTEL ASSEPON YEUREY TELTAL
 400   HA REISS DU UN MU MEA ANT HA REISS CODEM ON CLINVA HA REISS CODEM UN MEA RAIR
 600   HA REISS FU UN DU DEA ANT HA REISS COFED ON CLINPA HA REISS COFED UN DEA RAIR
 800   HA REISS FU UN TU TEA AND HA REISS COFET ON CLINGA HA REISS COFET UN TEA RAIR
1000   OE SHILL FU UN TU THE END OE SHILL CAFHT AN CRINGE OE SHILL CAFHT UN THE SEIS
1200   ME SHILL GO ON TO THE END ME SHILL PAGHT AN PRINCE ME SHILL PAGHT ON THE SEIS
1400   ME SHALL GO ON TO THE END ME SHALL WIGHT IN WRANCE ME SHALL WIGHT ON THE SEAS
1600   ME SHALL GO ON TO THE END ME SHALL WIGHT IN WRANCE ME SHALL WIGHT ON THE SEAS
1800   WE SHALL GO ON TO THE END WE SHALL PIGHT IN PRANCE WE SHALL PIGHT ON THE SEAS
2000   WE SHALL GO ON TO THE END WE SHALL FIGHT IN FRANCE WE SHALL FIGHT ON THE SEAS
```

Actually after 2000 iterations we have got the plaintext (with correct interpunctuation):

> *We shall go on to the end, we shall fight in France, we shall fight on the seas and oceans, we shall fight with growing confidence and growing strength in the air, we shall defend our Island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills; we shall never surrender, and even if, which I do not for a moment believe, this Island or a large part of it were subjugated and starving, then our Empire beyond the seas, armed and guarded by the British Fleet, would carry on the struggle, until, in God's good time, the New World, with all its power and might, steps forth to the rescue and the liberation of the old.*
>
> Winston S. Churchill, House of Commons, June 20, 1940

It is quite amazing how quickly and automatically the text was deciphered.

Many important question arise now, but we shall defer these to section 2.

11.1.6 The Vigenère Cipher, le chiffre indéchiffrable

To summarize our findings: monoalphabetic substitution ciphers can be broken routinely by frequency analysis of letters or bigrams. The major reason for this weakness is that a particular letter in plaintext is always mapped *to the same* latter in ciphertext. Thus there are always revealing footprints in the frequencies of letters in ciphertext. Provided the ciphertext is sufficiently long frequency counts give us statistically significant signals which can be used to break a cipher. So these systems cannot be considered secure.

This all was certainly known in the 15th century. Leon Battista Alberti (1404-1472), a remarkable Renaissance scholar working as poet, architect, painter and also as cryptographer was apparently the first to propose a cipher which (seemingly) rules out frequency analysis. His idea was to use more than one alphabet for encryption. By doing so, one and the same letter will have several equivalents in ciphertext. E.g., a plaintext `a` may be mapped into `W` on its first

occurrence. The next `a` may be mapped to `B`, etc. It is this idea which forms the basis of what is known as *polyalphabetic substitution*. Observe the intended effect of polyalphabeticity: it is to *flatten* the distribution of letter frequencies f_i as far as possible. There will be still peaks in the distribution but they are merely the result of the random variation of the statistical estimates f_i.

But, what ciphertext alphabets should be used?

How do we determine which alphabet has to be used in a particular step of the encryption process?

Several solutions have been proposed among these the classical Vigenère cipher by Blaise Vigenère in 1586[3].

Encryption and decryption

The Vigenère cipher uses a table of alphabets, called *tabula recta*, see Table 6. It is has 27 lines, the line d being our standard alphabet shifted *left* by $d = 0, 1, \ldots, 26$. Thus these are all Caesar's codes!

	_	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
_	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Table 11.6: The Vigenère table for our standard alphabet

Encryption is best explained by an example. In the sequel we assume a secret key k is used for communication and that the length of k is < 27, the size of

[3] Actually this cipher has been invented earlier by Giovanni Battista Bellaso in 1553.

the standard alphabet[4].

Example 6. Suppose we want to encipher the plaintext *meet you on sunday at nine* with key `PLAYFAIR`. First we form the *key string* by perpetuating the key until the key string has the length of the plaintext:

Key	`PLAYFAIRPLAYFAIRPLAYFAIRPL`
Plaintext	`meet you on sunday at nine`
Ciphertext	`BQFRFZXLP OYYVWVQJAZZAW CQ`

Then process each letter of the plaintext in turn:

- The first plaintext letter `m` has to be enciphered with the first letter of the key string `P`.
- For this purpose we use the alphabet in line `P` in the *tabula recta*.
- Using this alphabet the letter `m` is enciphered as `B`, etc.

Of course, using the Vigenère cipher this way is really awkward. As a result the application of the cipher for instance in the battle field was practically too difficult and cumbersome. But may be that people were not aware of *molular arithmetic*? Indeed, there is a very simple algebraic implementation close to that we used for Caesar's Cipher.

Let k denote the key string. For instance, in Example 6 we had

$$k = \texttt{PLAYFAIRPLAYFAIRPLAYFAIRPL},$$

and k_i the integer representation i-th letter of k which we obtain by applying Table 1. So $k_1 = 16, k_2 = 12$, etc. Furthermore, let a_i and c_i denote the integer representation of the i letter of the plaintext and the ciphertext. Then enciphering is most easily done by:

$$c_i = (a_i + k_i) \bmod 27, \qquad i = 1, 2, \ldots, n \tag{11.3}$$

where n is the length of the plaintext.

So:

`plain`	a_i	`key`	k_i	$a_i + k_i$	c_i	`cipher`
`m`	13	`P`	16	29	2	`B`
`e`	5	`L`	12	17	17	`Q`
`e`	5	`A`	1	6	6	`F`
`t`	20	`Y`	25	45	18	`R`

Decryption is done in the inverse way:

$$a_i = (c_i - k_i) \bmod 27, \qquad i = 1, 2, \ldots, n \tag{11.4}$$

Thus, *knowing the key* decryption is also very easy. But what, if we do not know the key?

[4]This assumption can be weakened considerably.

Cryptanalysis

We have already remarked that the secureness of a system is measured by the size of its key space $\mathcal{K}|$. Let d denote the length of the key. E. g., for `PLAYFAIR` we have $d = 8$. Then (allowing repeated letters in the key):

$$|\mathcal{K}| = 27^d, \quad \text{in our example: } |\mathcal{K}| = 27^8 \doteq 2.82 \cdot 10^{11}$$

This is much less than for the monoalphabetic substitution cipher. But we know, simple frequency analysis is now out of business. This is why Vigenère was also called *le chiffre indéchiffrable*.

Still, statistical methods can be used for a very effective attack. There are two severe weaknesses of Vigenère which can be used to break it:

- The key string is formed by repeating the key. This generates a *periodicity* which may leave its footprints in ciphertext.
- The alphabets are simple cyclic permutations of the standard alphabet. Thus once an alphabet is chosen the corresponding plaintext letter is enciphered using a simple Caesar's Cipher which we know is very easy to break.

By these observations it should be clear that finding the *length* of the key is the crucial point. Once known the rest of the business is done by frequency analysis as we have it outlined in Section 1.3.

There are various approaches to find the key length d. One is based on the use of the *index of coincidence* $\Phi(T)$ invented by William F. Friedman[5].

Let T be a text over *some* alphabet. Assume that the length of T is $|T| = n$ and the alphabet consists of N letters. Then $\Phi(T)$ is an estimate of the probability that two randomly chosen letters in T are the same:

$$\Phi(T) = \frac{1}{N(N-1)} \sum_{i=1}^{n} F_i(F_i - 1),$$

where F_i denotes the *absolute frequency* of the i-th letter of the alphabet in the text T. For English text the value of $\Phi(T)$ is around 0.07, whereas for random text $\Phi(T) = 1/27 = 0.037$, all letters being equally probable.

Now let $T = C$, C being the cipher text and suppose the key has length d. Then we split C into d blocks:

$$\begin{aligned} C_0 &= [c_0, c_d, c_{2d}, \ldots] \\ C_1 &= [c_1, c_{d+1}, c_{2d+1}, \ldots] \\ C_2 &= [c_2, c_{d+2}, c_{2d+2}, \ldots] \\ &\ldots \\ C_{d-1} &= [c_{d-1}, c_{2d-1}, \ldots] \end{aligned}$$

[5] William Frederick Friedman (1891 - 1969) was one of the greatest cryptologist of all time.

If the key length is indeed d, then these blocks should look like English text for $\Phi(C_i)$. So we calculate the indices of coincidence for each group C_i and take the average:

$$\Phi(C) = \frac{1}{d}\left[\Phi(C_0) + \Phi(C_1) + \ldots + \Phi(C_{d-1}\right].$$

The value of $\Phi(C)$ should be around 0.07 if the key length is indeed d, otherwise $\Phi(C)$ will be much smaller.

Let's try this.

Example 7. Suppose we have intercepted the following message which we know (from some source) is enciphered using the Vigenère system and plaintext language is English.

```
GWGMRUVTMZSSATENRIBDRUFGNHUZRKODNHWTSOTIMFEBZZEUNHAUSEXI
ENRKODNHWTSONGWSWRNHRSS NN PAGMVEENHAUGSSGRLMFNP YRQECZI
KFNHUZRQSPPZRPTOFBWACPPCXJQOFBIPHVUUUPBDRLSUWC UNJGWMNZF
NPZZIJQP UYPITDHDFAHMBSTNUVHSMZMMWWFAOYIJUNHUZRKODNHWTSO
TIMFEBZZEUNFRAIFGGMNFAVPHZRUBO IKJTMMBWSSQKUKISONGWSWRNH
RHBJRLENSBFUKIOHMCEAIXRQRPTOFBWAOHFCKVRTMIXAGWRUSNSFVXSO
NVAPWSAARHKAWHMXSOACFUTVGOPIETWSRLRUVPFU UNXEU NCCEM CZT
MNFAETNXZAOBMVYSSTZZEUNHULFVUWM LSGWRLROSVAN BGXAHJ
```

For this text C I have calculated $\Phi(C)$ for conjectured key lengths $d = 1, 2, \ldots, 25$. The results are given in Figure 3. You can see the striking peaks at $6, 12, 18, 24$. This is a strong indication that the unknown key has length $d = 6$.

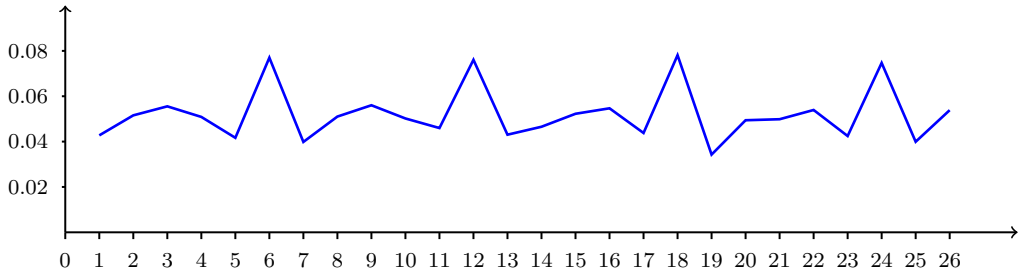


Figure 11.3: The index of coincidence for the cryptogram in Example 7

Once we have a good guess of the key length d the rest is easy. Since all alphabets of the Vigenère table are shifted Caesar's alphabets, just apply the technique outlined in Section 1.3 to determine the shift by means of total variation distance.

Having done so, we find that the secret key is `NOMURA` and the plaintext of the cipher is (with correct interpunctuation):

```
Thus, the earnest hope of the Japanese Government to adjust
Japanese-American relations and to preserve and promote the peace
```

```
of the Pacific through cooperation with the American government
has finally been lost.

The Japanese Government regrets to have to notify hereby the
American Government that in view of the attitude of the American
Government it cannot but consider that it is impossible to reach
an agreement through further negotiations.
```

This is indeed a famous document dating from December 7, 1941. It is the last page of the Japanese note handed to Secretary of State Cordell Hull while Pearl Harbor was being attacked by Japanese forces. Nomura was the name of the Japanese ambassador at Washington. The thrilling story about this cryptogram is told in Chapter 1: *A Day of Magic* of David Kahn's book (Kahn, 1996).

11.1.7 Transposition Ciphers

Encryption and decryption

We already talked briefly about transposition ciphers. These come in an impressing number of variants. The basic feature of these systems is that letters retain their value but change place in text.

Here I will describe only the simplest system.

The correspondents agree upon a secret key p which is a permutation of the numbers $1, 2, \ldots, d$ for some $d > 1$. The plaintext is divided into blocks of length d and letters *within* each block are permuted according to p.

Example 8. The plaintext is `troops gathering attack from north` and this should be enciphered with key $p = [4\ 1\ 5\ 3\ 2]$. Thus $d = 5$. Encryption goes this way (for reasons of readability the space character is printed as underscore letter):

key p	41532	41532	41532	41532	41532	41532	41532
plaintext	troop	s_gat	herin	g_att	ack_f	rom_n	orth_
ciphertext	OTPOR	ASTG_	IHNRE	TGTA_	_AFKC	_RNMO	HO_TR

Decryption is also easy, just take the *inverse permutation* p^{-1} and recover the plaintext from the ciphertext.

Example 8. (continued) The inverse permutation is found easily by writing p as 2-rowed array, sorting the second row and exchanging rows:

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 5 & 3 & 2 \end{pmatrix} \Longrightarrow \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 4 & 1 & 3 \end{pmatrix} = p^{-1}$$

Hence:

inverse p^{-1}	25413	25413	25413	25413	25413	25413	25413
ciphertext	OTPOR	ASTG_	IHNRE	TGTA_	_AFKC	_RNMO	HO_TR
plaintext	troop	s_gat	herin	g_att	ack_f	rom_n	orth_

Cryptanalysis

Transposition ciphers are just what cryptanalysts are waiting for, as was reported e.g. about the German *Abwehr* during World War II. The key space has size $d!$ which is quite small unless d is large. In Example 8 we have $5! = 120$, thus a brute-force attack doesn't result in serious computational trouble. Frequency analysis, however, seems to be out of business now. Though ..., see Section 2 for more about this and related questions.

11.1.8 Perfect Secrecy

After having been introduced to some classical ciphers and after you have seen that these can be broken rather routinely you may wonder whether there exists a cipher system that cannot be broken, a system guaranteeing *perfect secrecy*.

Here is a naive argument which tells us: No, there can't be perfect secrecy, because

- All messages we send and receive have finite length, they consist of a finite number of symbols, thus require finite time for transmission.
- Therefore it is always possible to find the plaintext of a ciphertext by brute force. It's just a matter of time and computing power. But, of course, it make take quite some time.

All right, but still there is a fatal flaw in this argument which I will demonstrate drastically in Example 9 below.

Actually, there are cipher systems giving us perfect secrecy. In informal terms: *a cipher system has perfect secrecy, if the unauthorized eavesdropper learns nothing about the plaintext from the ciphertext.*

This informal statement can be made strict by means of conditional probability. Perfect secrecy has been defined and thoroughly discussed by Claude Shannon (1948) and (1949) in his seminal papers.

An example of a system having this remarkable property is *Vernam's Cipher*. It is breathtakingly simple!

We will apply an algebraic representation as we have used it with Caesar's and Vigenère's cipher. The basic ingredient of Vernam's cipher is the key:

- it must have length equal to the length of the plaintext;
- it must be absolutely random.

By Table 1, there is a one-to-one correspondence between letters of the standard alphabet and the integers $0, 1, \ldots, 26$. Again, let a_i denote the numerical value of the plaintext letter at position i, k_i the value of the key letter and c_i that of the ciphertext letter. Assume that the plaintext has length n. Then we have:

$$\begin{aligned} &\text{encryption:} \quad c_i = (a_i + k_i) \bmod 27, \qquad i = 1, 2, \ldots, n \\ &\text{decryption:} \quad a_i = (c_i - k_i) \bmod 27 \end{aligned}$$

Example 9. In the following table a plaintext and a key of equal length are given and enciphered:

plaintext	`enemy will surrender tomorrow`
key	`TQPLNFDZYADMNQS SP RUSJYDTOFP`
ciphertext	`YDUYLF HJMDEHHJEFTEIULYKSKFUL`

For the first two letters:

$$a_1 = \texttt{e} \mathrel{\widehat{=}} 5, \quad k_1 = \texttt{T} \mathrel{\widehat{=}} 20, \quad c_1 = (5 + 20) \bmod 27 = 25 \mathrel{\widehat{=}} \texttt{Y}$$
$$a_2 = \texttt{n} \mathrel{\widehat{=}} 14, \quad k_2 = \texttt{Q} \mathrel{\widehat{=}} 17, \quad c_2 = (14 + 17) \bmod 27 = 4 \mathrel{\widehat{=}} \texttt{D}$$

What about *cryptanalysis*? Let us assume that the cryptanalyst knows that this is a Vernam cipher but he does not know the key.

Indeed, the cryptanalyst is in a very weak position now. The plaintext has length $n = 29$ which means that there are 27^{29} keys to be considered in the worst case if a brute-force attack is run. But:

$$27^{29} = 32325790992917453429227398072136027185338 7 \simeq 3 \cdot 10^{41}$$

Of course, only a relatively small percentage of keys will yields sensible English text. Eventually the cryptanalyst may find the correct key. But during key search with some positive probability he may also come across the key `LCHXLJRWYMTWWWJPEDDIAZLGATRBL`, which (alas!) yields:

ciphertext	`YDUYLF HJMDEHHJEFTEIUNYVSKFYL`
key	`LCHXLJRWYMTWWWJPEDDIAZLGATRBL`
plaintext	`mama will kill papa tomorrow`

In other words, an exhaustive key search will yield all sensible English[6] text of a given length. Thus the cryptanalyst runs into a difficult decision problem which can hardly be resolved! That's why Vernam's cipher is perfect. Yet it is not foolproof!

11.2 Where to go from here

After having read this *Invitation* so far (about 20 pages!) you may wonder whether there is anything left to do for you?

There remains quite a lot of work to be done.

Write an interesting and exciting thesis about elementary methods of cryptology. Your paper should be a nice mix of theoretical considerations, historical notes and, of course, it should also have a computational flavor. Your thesis should also contain several examples to demonstrate your findings.

[6] of course also German, French, Russian,...

11.2.1 Issues of general interest

- At the outset you should make up your mind what programming language you will use. For instance, all examples in this introduction are written in C. But you are free to use any other language like Java, R, etc. It may also be helpful to use some scripting language like perl.

- Next organize an appropriate text corpus to have a learning sample. The texts may be in English or German and should be in total sufficiently long (about 3 mega bytes, or so). Based on this learning sample:
  - Perform a careful statistical analysis of these texts.
  - Determine frequencies of letters, bigrams, may be also of trigrams (sequences of three contiguous letters).

  Regarding the text corpus: you may use English or German texts, but take care of copyright protections.

- Implement an automatic decryption routine for Caesar's cipher. You may or may not use total variation distance. There is also a quadratic measure which will remind you in the χ^2-statistic.

- Give a careful discussion of the Metropolis algorithm. It is actually a special case of a *meta heuristic* known as *Simulated Annealing*. SA is capable of more, the driving master process of SA is able to intensify and diversify search.

- Implement the Metropolis algorithm to break a monoalphabetic cipher. Try to be as general as possible so that your implementation can be easily reused to solve other, harder problems.

- Give a thorough discussion of the classical Vigenère cipher and implement a routine which can break this system.

- Discuss in detail Friedman's index of coincidence $\Phi(T)$ and related measures.

- Implement a routine to solve the simple transposition cipher introduced in Section 1.7.

- I have remarked that the Vernam cipher is not foolproof. It can be broken if not used properly. Give a careful discussion of the conditions for proper use of this cipher.

11.2.2 Some more suggestions

- When playing and experimenting with the Metropolis algorithm, for instance, you will find out, that to break a cipher you will need a minimum amount of ciphertext available, the more, the better. Indeed, there is a minimum length of ciphertext needed to guarantee a *unique* decryption. This length is known as *unicity distance* U and it is closely related to the

concepts of *entropy* and *redundancy*. For simple substitution ciphers U is surprisingly small, about $U = 30$ for English text. However, for the Vernam cipher, $U = \infty$. Discuss U.

- Recall that the fatal weakness of Vigenère's cipher is periodicity generated by the key, and this can hardly be hidden. We have already discussed finding the key length by means of the index of coincidence. But there is another famous method: *Kasiski's Test*[7], which tries to identify recurrent patterns in the ciphertext and deduce thereby the length of the secret key. Discuss and implement Kasiski's test.

- Once the length of the key is known the classical Vigenère cipher is rather straightforward to break because the alphabets used are simple shifted Caesar's alphabets. But the strength of Vigenère can be boosted considerably when the Vigenère table consists of (in our cases 27) different *random permutations*. The size of the key space is thereby increased from 27^d to $(27!)^d$ where d is the length of the key. Devise an algorithm to break this general Vigenère cipher. Metropolis may be helpful in this context.

- Actually, using more general alphabets in the Vigenère table is an old idea already suggested by Porta[8]. Vigenère actually invented another system to generate the secret key, the *autokey cipher*. This method uses the plaintext to become part of the key. Discuss the idea of autokey and, if possible, implement a method to break the resulting cipher.

- So far we have only seen simple transposition ciphers, but there is an incredible number of variants. You may also discuss one or the other example of more exotic transposition systems.

- An interesting questions is: how can the cryptanalyst find out *what* cipher is used? Are there methods to identify the cipher system?

11.2.3 What to be avoided

Your thesis should cover basic cryptology up to 1918, the end of World War I. THis is a key date, as early in 1918 Arthur Scherbius patented the first electromechanical cipher machine based an rotors, the *Enigma Machine* and this initiated a new era in cryptology.

I would appreciate if you avoid discussing ciphers like Enigma and its various descendants, the *Data Encryption Standard (DES)* or the *Advanced Encryption Standard (AES)* which is widely used today. Also ciphers based on number theory like RSA should not be topic in your thesis. All these are very advanced systems requiring special mathematical methods that I cannot afford from you.

[7] Friedrich Wilhelm Kasiski (1805 - 1881) was a prussian infantry officer. In 1863 he published a small booklet on cryptology which became on of the most influential and important works in this field.

[8] Giambattista della Porta, 1535-1615, Renaissance scholar.

Please do not be disappointed about this restriction.

Now, after having worked out the major issues there remains on final job for me to be done: *Enjoy writing this thesis! Have fun!*

11.3 An Annotated Bibliography

The book Kahn (1996) is *the classical text* about the history of cryptology. This is really an exciting book covering the field from ancient times up to the end of the 20th century. Chapter 1, *One Day of Magic* is the thrilling story of American codebreakers around William Friedman and the Japanese attack on Pearl Harbor in 1941. The crucial Japanese diplomatic notes (we have seen the last one in Section 1.6.2) were, of course, not encrypted in Vigenère. Japan used several much stronger systems, practically all broken by the United States Signal Intelligence Service to which William Friedman belonged.

Bauer (2007) is an excellent introduction to cryptography and cryptanalysis. The book has two parts. In the first part standard methods of cryptography are introduced. The second part is devoted to cryptanalysis. The book is full of interesting examples. Part II gives a fairly complete coverage of the most important statistical methods for cryptanalysis. No special mathematical knowledge is required to read and understand this book except for some basic terms like relations and functions and the corresponding mathematical notation. There is also a German edition.

Modern cryptanalysis by Swenson, 2008 is another remarkable textbook on the subject, however, only Chapter 1 will be relevant for your thesis.

You will also enjoy the booklet by Gains (1956). Chapters 1-7 are devoted to transposition ciphers, chapters 8-23 to substitution ciphers. It contains many solved examples and you will find here also a thorough discussion of polyalphabeticity, in particular of the Kasiski Test.

The paper Diaconis (2008) discusses the Metropolis algorithm for breaking monoalphabetic ciphers. Only the first few pages will be interesting for you because there you find a description of the Metropolis algorithm and a plausibility measure. The major part of this article deals with convergence problems and representation theory of finite groups. Chen and Rosenthal (2010) is a technical report, you will find very interesting. It introduces the Metropolis algorithm with an alternative plausibility measure and discusses applications to various ciphers including simple transposition. Also, the authors report some statistics recorded in their experiments with different text corpora and different choices of parameters. An abridged version having been published (Chen and Rosenthal, 2012) in *Statistics and Computing*.

11.4 References

[1] Friedrich L. Bauer. *Decrypted Secrets, Methods and Maxims of Cryptology*. Springer, 2007.

[2] Jian Chen and Jeffrey S. Rosenthal. *Decrypting classical cipher text using Markov chain Monte Carlo*. 2010. URL: http://probability.ca/jeff/ftpdir/decipherart.pdf.

[3] Jian Chen and Jeffrey S. Rosenthal. "Decrypting classical cipher text using Markov chain Monte Carlo". In: *Statistics and Computing* 22.2 (2012), pp. 397–413.

[4] Persi Diaconis. "The Markov chain Monte Carlo revolution". In: *Bulletin of the American Mathematical Society* 2 (2008), pp. 179–205.

[5] Helen Fouché Gains. *Cryptanalysis, a Study of Ciphers and Their Solution*. Dover Publications, 1956.

[6] L. Graham Ronald, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. 2nd ed. Addison-Wesly, 2003.

[7] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, New York, 1996.

[8] W. W. Rouse Ball and H. S. M. Coxeter. *Mathematical Recreations and Essays*. 13th. Dover Publications, 1987.

[9] Claude E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[10] Christopher Swenson. *Modern Cryptanalysis: Techniques for advanced Code Breaking*. Wiley Publications, 2008.

Topic 12

Parrondo's Paradox
The Gambler's Perpetuum Mobile

Keywords: *probability theory, fair games, Markov chains*

12.1 An Invitation

12.1.1 Favorable and unfavorable games

Gambling lies at the roots of probability theory. From its very beginning as a mathematical discipline in the 16th and 17th century problems arising from studies of various games have been impetus to many important and fruitful developments. Just to mention the classical gambler's ruin problem, the Central Limit Theorem or even martingale theory. Important contributors at these early days have been Fermat, Huygens, Pascal, De Moivre and J. Bernoulli. Various paradoxes ever and ever turned up and initiated lively disputes, e.g. De Méré's Paradox, which is trivial to resolve, or the St. Petersburg Paradox, which is not so trivial.

Parrondo's Paradox is a relatively new one, its first version has been published 1997. It is this paradox which your thesis should be about.

Consider a two-person game, player X plays against player Y. The game consists of an arbitrary number of rounds or single plays in which the event that X or Y wins is determined by the outcome of a random experiment. This could be e.g., tossing a coin, throwing a dice, spinning a roulette wheel, etc. Let p denote the probability that X wins a single game. If $p > 1/2$, we say that the game is *favorable* for X. On the other hand, if $p < 1/2$ the game is *unfavorable* for X.

If $p = 1/2$, we are tempted to say that the game is *fair*. However, one should be rather cautious with the term *fair*. The random experiments constituting the game need not be independent. And even if they were, it is possible to construct games with success probability $p = 1/2$, where in a prolonged sequence of games a player will be on the losing side. This may be the case, for instance, if the payoff of a single game is a random variable with infinite variance.

Despite of this warning don't let us be overly pedantic and call a game fair, if $p = 1/2$. It is a quite remarkable fact that there exists no winning strategy

in a sequence of independent games with success probability $p = 1/2$. This is by no means trivial and you should also elobarate on this point in your thesis. Chapter X.3 of reference [2] will be a good source.

12.1.2 Combining strategies

So, no winning strategy in case $p = 1/2$, certainly no one necessarily if $p > 1/2$. Can be hope that there is one for $p < 1/2$? Surely no, you may think. But now Parrondo's Paradox enters the scene!

Consider the following game, let's call it game A: it is played on a roulette table, the wheel has slots for 0 and the numbers $1 \ldots 36$. The zero is colored green, among the other numbers half is colored black, the other half colored red. All numbers are equally likely. When zero turns up the casino always wins. The gambler's capital consists of a number of chips, each one € worth. The somewhat strange rules of game A are these:

Game A If your capital is a multiple of 3 and one of the numbers 1,2 or 3 turns up, you win one €. This happens with probability 3/37. Otherwise, if your capital is not a multiple of 3, you win one €, if the outcome is a number in the range $1, 2, \ldots, 28$, which will happen with probability 28/37.

In all other cases you lose.

Is this a favorable game for you?

A naive argument leads to the conclusion that it is. Here is the argument: in any round your capital is a multiple of 3 with probability 1/3 and with probability 2/3 is not a multiple of 3. Hence the probability of winning should be

$$p = \frac{1}{3} \cdot \frac{3}{37} + \frac{2}{3} \cdot \frac{28}{37} = \frac{59}{111} \doteq 0.532 > \frac{1}{2}$$

So, that looks fine! But beware, there is something seriously wrong with this argument, indeed, game A favors the casino, as actually the winning probability equals $p = 0.494 < 1/2$!

Why that? Find it out! You will have to learn a little bit about finite Markov chains to answer this question, see reference [3].

So far, so bad. So, let's consider another game, say game B. Its rules are very simple:

Game B You bet on red or black and win one €, if that color turns up. This happens with probability 18/37.

Again this game is favorable for the casino.

But now, Parrondo's Paradox: suppose that any time a new round starts, you have the *choice* to play game A or game B. Your decision which game to play is completely free. It may depend on the outcomes of previous games, you may play the games in a purely deterministic pattern, like $AABAABAAB\ldots$ or $BABABAB\ldots$. You may even toss a coin and play A, if head turns up and play game B, if tail comes. It's a remarkable and really stunning fact that the combination to two unfavorable games is thereby turned into a favorable game! Even if you play A and B in random order your winning probability will be $p = 0.502 > 1/2$! How is this possible? Find it out.

So it seems that there is money for free! That's why the subtitle of your thesis is *the gambler's perpetuum mobile*. But there is another reason for this subtitle. Juan Parrondo invented this game for pedagogical reasons to propagate his idea of a *Brownian Motor*, a very strange effect in thermodynamics.

12.2 Where to go from here

12.2.1 Issues of general interest

- Discuss the notion of a fair game and the nonexistence of gambling systems in independent sequences of identical games.
- Simulate with a computer, e.g. in **R**, various patterns of games A and B, like

  $$ABABABABA\ldots, AABAABAAB\ldots, BBBABBBBAB\ldots$$

  and determine empirically the winning probabilities for these patterns. Can you find a pattern which maximizes this probability?
- Discuss potential applications of Parrondo games.

12.2.2 Some more suggestions

- Calculate the exact values of the winning probabilities for various patterns.
- In references [1] and [4] you will find a canoncical formulation of Parrondo games which contains an *unfairness parameter* ϵ. Perform numerical experiments to explore the effect of this parameter.

Note. Your own ideas and creativity are always welcome!

Feller (1970)

12.3 An Annotated Bibliography

Still missing ...

12.4 References

[1] William Feller. *An Introduction to Probability Theory and Its Applications*. 3rd. Vol. 1. John Wiley and Sons, 1970.

[2] G. P. Harmer and D. Abbot. "Parrondo's Paradox". In: *Statistical Science* 14.2 (1999), pp. 206–213.

[3] D. Minor. "Parrondo's Paradox: Hope for Losers!" In: *The College Mathematics Journal* 34.1 (2003), pp. 15–20.

Topic 13

Runs in Random Sequences

Keywords: *run statistics, nonparametric statistics, extreme value distribution, elementary renewal theory, generating functions*

This chapter is has not been finished yet.
October 4, 2018

13.1 An Invitation

13.1.1 Some remarkable examples

Instead of beginning this description with rather technical definitions let me present three examples which should give you an idea of what this thesis should be about.

Example 1. In the evening of August 18, 1913, in the famous casino of Monte Carlo the color black turned up 12 times in a row on a roulette table. After the 12th occurrence of black more and more people gathered around that table and began to set their stakes on red, as most of them believed that this color was heavily overdue. But black continued to turn up a 13th time, a 14th time. Stakes were increased, but all those betting on red lost. Indeed this run of black had finally a length of 26! Many people lost a lot of money and rumors said that never before the Casino made such nice profits during one evening only.

Example 2. The world chess championships in the 1970s and 1980s were played in an atmosphere of almost paranoic suspicion. A paramount example of this paranoia was the championship 1985, Kasparov versus Karpov. Bobby Fisher, world champion 1972, claimed that the matches between Kasparov and Karpov were rigged and prearranged move by move. Fisher asserted 1996 that in a particular match starting with move 21 White makes no less that 18 consecutive moves on the light squares! Is this run an evidence for Fisher's claim?

Example 3. This example is usually attributed to P. Révész (former professor at TU Vienna), but Révész once told me that actually Prof. Tamas Varga was the first to perform the following experiment. When a professor of probability entered his class for the first lecture he asked his students to form two groups

of equal size. In the first group each member should toss a coin and write down a zero if head comes and a one if tail occurs until a sequence of length 100 is complete. The students of the second group should *simulate* a sequence of zeroes and ones of length 100, i.e. they should write down what they believed a random sequence should look like. The professor then left the class and return 15 minutes later. The sheets with the sequences have been collected and shuffled. He accepted bets that he could distinguish simulated from true random sequences with high probability. Actually, he almost always won, indeed, his success rate was more than 90 %.

Here are two typical sequences:

Sequence 1:

```
01111000011110111111110001100111110000011010010001
11011000010111011100111100000011100110001000011110
```

Sequence 2:

```
10101110111110100111011000100101011110011001100010 1
00011000111000110000111019100000111000100101110001 0
```

Can you see which of the two is random and which is simulated?

These examples have something very interesting in common:

- We have a sample of dichotomous observations, red and black, light and dark squares, zeroes and ones.
- The samples contain *runs* of various lengths. A run being an unbroken sequence of the same data point.
- In examples 1 and 2 we are observing very long runs. In these cases there is reasonable suspicion that the observed sequences show some *systematic pattern*, i.e., they are *non-random*.
- In example 3, T. Varga performs *a statistical test on randomness*, his test statistic being the length of the longest run observed in the sample. And it seems that this test statistic is a very good one, because otherwise he would not risk his money.

These examples give rise to several interesting questions, in particular this one:

> *Given a sequence of observations, is this sequence purely random or does it show systematic patterns?*

This question is of considerable importance, e.g.,

- Checking the quality of random number generators which are an essential part of simulation studies;
- In data encryption to check the security of a cipher;
- In nonparametric statistics: the test of Wald and Wolfowitz is based on runs to test the hypothesis that two samples come from the same population.
- In computer science this is just a special case of the more general pattern matching problem.

13.1.2 Important random variables related to runs

Let us now be a little bit more specific: consider a 2-letter alphabet $\mathcal{A} = \{0, 1\}$. Let S_n be a string of length n formed over the alphabet $\mathcal{A}$. Here are some examples of strings of lengths $n = 20$:

$$S_1 = \texttt{00111100111000001011} \quad S_2 = \texttt{11111111110000000000}$$
$$S_3 = \texttt{01010101010101010101} \quad S_4 = \texttt{11101100000010111011}$$

A 1-run is defined as an unbroken sequence of 1s of particular length, similarly one may define a 0-run. For instance, S_1 has 1-runs of lengths 1, 2, 3 and 4. S_2 has one 0-run of length 10 and one 1-run of length 10.

Let us now assume that the sequences are generated by independent Bernoulli-experiments, i.e. experiments which have exactly two outcomes 1 and 0 with probabilities p and $q = 1 - p$.

The following random variables are of interest for random sequences of length n:

$$\begin{aligned} N_n(k) &= \text{ the number of 1-runs of length } k \\ T_n(k) &= \text{ the position of first occurrence of a 1-run of length } k \\ R_n &= \text{ the length of the longest 1-run} \\ U_{n,i}(k) &= \text{ the indicator of 1-run of length } k \end{aligned}$$

The indicator $U_n(k)$ is defined as

$$U_{n,i}(k) = \begin{cases} 1 & \text{if a 1-run of length } k \text{ occurs at position } i \\ 0 & \text{otherwise} \end{cases}$$

Here is an important point regarding these random variables: whenever a 1-run of length k is completed, the recording starts from scratch. Thus we assume that runs are *non-overlapping*. This makes the analysis easier, but of course, more general definitions are possible and indeed in use.

Just to give you an example, let

$$S_{30} = \texttt{010111110100011110111111101100}$$

and consider 1-runs of length $k = 4$.

Here, $T_{30}(4) = 7$, because the first 1-runs of length 4 completes at the 7-th trial. Then recording starts anew, so that the next run of length 4 completes at position 21 and not in position 8!

Observe also that in this example $U_{30,7}(4) = 1$, but $U_{30,8}(4) = 0$, furthermore $R_{30} = 7$.

There are several important connections between these random variables, you should find out which they are!

The most interesting of these random variables is R_n, the length of the longest run. It can be shown that for large n

$$R_n \approx \frac{\ln(nq)}{\ln(1/p)}$$

Here $\approx$ means *is close to*. This is the famous Erdös-Renyi-Law. For instance, if $n = 100$, and $p = 1/2$, then

$$R_{100} \approx \frac{\ln 50}{\ln 2} \doteq 5.64$$

But much more can be said, in particular, and this is really amazing, the variance of R_n is practically independent of n. Thus it is possible to *predict* R_n with high accuracy!

13.1.3 Methodological Issues

There are several ways to attack these problems:

- Elementary renewal theory is the easiest way and therefore strongly recommended. A good introduction may be found in reference [1], chapter XIII.
- An alternative is the use of finite Markov chains, see reference [4]. For an elementary exposition of finite Markov chains reference [5] is recommended.
- The most powerful approach is through symbolic methods and generating functions, as is mostly done in computer science. This is technically considerably more demanding but also much more flexible, you will find a lot of interesting material in reference [3].

13.2 Where to go from here

13.2.1 Issues of general interest

- Give a careful analysis of the random variables defined above and show how they are connected.
- Give examples and discuss interesting applications of run statistics. The test of Wald and Wolfowitz would be such an application. It is often used as a statistical test on randomness, but it is also used for the classical two sample problem: there are two samples $\mathbf{X}$ and $\mathbf{Y}$ with sample sizes m and n, and continuous distributions $F(x)$ and $G(x)$, respectively. The null hypothesis is: the samples come from the same population, i.e., $H_0 : F = G$. An alternative might be: $H_A : F \neq G$. Reference [8] may be helpful, in particular chapter 11 and chapter 12.6 of this book.
- Perform numerical experiments, e.g. using **R**, to estimate the probability distributions of these random variables. What do you observe when n becomes large? What happens to the distribution of $N_n(k)$, to the distribution of R_n when n becomes large? You will find, that there is a marked difference!
- Find out how the test Prof Varga used works, calculate exact p-values and test which of the two sequences given in example 3 is random.

13.2.2 Some more suggestions

The distribution of the longest runs R_n is a typical *extreme value distribution*. Discuss this important class of distributions and their applications in statistics, insurance, material testing, etc.

This chapter needs a lot of more work from my side ...

Note. Your own ideas and creativity are always welcome!

References.

1. Feller W. (1970), *An Introduction to Probability Theory and its Applications*, 3rd ed., John Wiley and Sons.

2. Fisz M. (1980), *Probability Theory and Mathematical Statistics*, Krieger Publishing. An edition in German language is also available.

3. Flajolet P., Sedgewick R. (2009), *Analytic Combinatorics*, Cambridge University Press New York.

4. Fu J. C., Koutras M. V. (1994), Distribution Theory of Runs: A Markov Chain Approach. *Journal of the American Statistical Association*, 89, 427, 1050–1058.

5. Kemeny J. G., Snell J. L. (1976), *Finite Markov Chains*, Springer, New York.

6. Schilling M. (1990), The Longest Run of Heads, *The College Mathematics Journal*, 21, 3, 196–207.

7. Schilling M. (1994), Long Run Predictions, *Math Horizons*, 1, 2, 10–12.

8. Segal M. R. (2007), Chess, Chance and Conspiracy, *Statistical Science*, 22, 1, 98–108.

Topic 14

The Myriad Ways of Sorting

Keywords: *Computer science, analysis of algorithms, divide and conquer algorithms*

14.1 An Invitation

14.1.1 Some basic terminology

In this thesis you should study a topic that arises very frequently in everyday life, in combinatorial mathematics and notably in computer science: the rearrangement of items into ascending or descending order. Imagine how hard it would be to use a dictionary if its words were not alphabetized. Or, if entries in the phone directory of your smart phone are not in alphabetic order.

Besides convenience sorting has many important applications: one is the *matching problem*. Suppose we have two long lists of names, say, and we want to pick out those names which occur in both lists. An efficient way so solve this problem is to sort both lists into some order. Then only one sequential pass is needed to identify the matching entries.

Another application is *searching* items in lists. Sorting makes the search process much easier. Indeed, many algorithms in computer science can be made considerably more efficient if input data are sorted.

And last, but not least: sorting techniques provide excellent illustrations of the general ideas involved in the *analysis of algorithms*, one of the most interesting areas in computer science.

Let us fix some terminology first. The items to be sorted are usually called *records* $R_1, R_2, \ldots, R_n$. The entire collection of records is a *file*. With each record R_i we associate a *key* K_j which governs the sorting process. For example, consider a phone directory. A typical key could be the surname of a person and with this key we associate *satellite information*, like phone number(s), address, and so on.

An *ordering relation* "$<$" on the set of keys must specified so that for each pair of keys a, b only one of the following can be true:

$$a < b \quad \text{or} \quad a > b \quad \text{or} \quad a = b.$$

Furthermore this ordering relation must be *transitive*. For any triple a, b, c of keys:

$$a < b \quad \text{and} \quad b < c \implies a < c.$$

In your thesis you should assume that keys are stored in an *array* in main memory of the computer, so that random access is possible. In other words, there is an array $\mathbf{a} = [a_1, a_2, \ldots, a_n]$ whose elements can be addressed by simple indexing. Thus for given $i, 1 \leq i \leq n$ we can access the key a_i in constant time. Sorting algorithms for this type of data are called *internal*. If data files are so huge that not all records fit into main memory, then sorting must be done *externally*. However, external sorting should not be in the focus of your thesis.

One more point: the process of sorting should be based on *comparisons* of keys. This means that there is a function, say `compare(a,b)`, with:

$$\texttt{compare(a,b)} = \begin{cases} 1 & \text{if } a > b \\ 0 & \text{if } a = b \\ -1 & \text{if } a < b \end{cases}$$

There are other sorting methods which are based on the representation of keys in a particular number system, e.g., the binary system. These are known as *radix methods* and should not be a major topic in your thesis.

It's time for an example to see how sorting can be accomplished. Suppose, we have $n = 10$ keys whose values are integers a_i:

i	1	2	3	4	5	6	7	8	9	10
a_i	2	9	6	8	10	1	7	4	3	5

Suppose also that we want to put the keys into *ascending order*.

14.1.2 An example: selection sort

One of the simplest sorting algorithms is *selection sort*. It works as follows: First, find the smallest item and exchange it with the first entry of the array. Then, look for the second smallest item and exchange it with a_2. Continue in this way until all keys are sorted. In Figur 14.1 the process of *selection sort* is illustrated. Thus in a few steps we finally find the key ordering:

$$[a_6, a_1, a_9, a_8, a_{10}, a_3, a_7, a_4, a_2, a_5] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].$$

Is selection sort a good algorithm?

A primary criterion to qualify a sorting algorithm as good is *efficiency*. The latter is usually measured by the number of comparisons C_n and the number exchanges E_n. For *selection sort* it is particularly easy to determine C_n and E_n. To find the smallest item $n - 1$ comparisons are necessary. We also need one exchange to move the smallest item to the first position. For the second

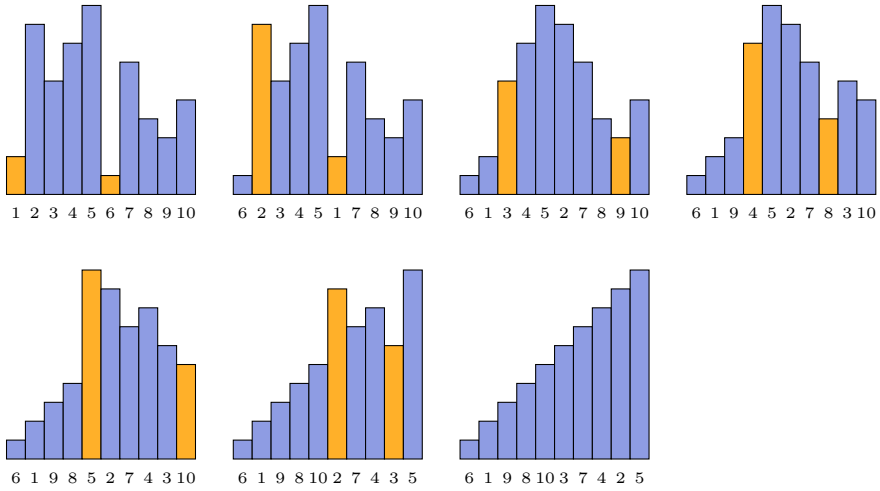


Figure 14.1: Selection Sort

smallest item we use $n-2$ comparisons and one more exchange, so

$$C_n = (n-1) + (n-2) + \ldots + 2 + 1 = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$$
$$E_n = 1 + 1 + \ldots + 1 = n$$

This is an interesting result, as it tells us: the number of comparisons C_n grows quadratically in n, the number of keys to be sorted. Thus doubling the number of keys increases the amount of computational work by a factor 4. One also says that *selection sort* has *quadratic time complexity*.

Selection sort is just one example of sorting algorithms with quadratic complexity, you should discuss other prominent algorithms in this class.

Algorithms with quadratic complexity are usually very simple, so it is easy to implement them in some programming language. They work perfectly well as long as the number of keys n does not become too large, say, $n < 1000$.

But what, if we have to sort a file with several hundred thousand of keys? This is by no means an exotic task, think of data files of social insurance institutions, for instance. For such large data files simple quadratic methods are no longer a viable alternative. Other ideas are needed.

14.1.3 Merging

Consider the problem of *two-way merging*. Given two already *sorted arrays* $\mathbf{x}$ and $\mathbf{y}$ we want to construct a new array $\mathbf{z}$ by merging $\mathbf{x}$ and $\mathbf{y}$ in such a way that $\mathbf{z}$ is also sorted. This is easy. We just compare to the smallest item, i.e. the first item in $\mathbf{x}$ with the first item in $\mathbf{y}$ and output the smaller one, remove it from $\mathbf{x}$ or $\mathbf{y}$ and repeat the same process until we are finished. For instance,

starting with

$$\begin{cases} \mathbf{x}: & 503 & 703 & 765 \\ \mathbf{y}: & 87 & 512 & 677 \end{cases}$$

we obtain

$$\mathbf{z}: \quad 87 \qquad \begin{cases} \mathbf{x}: & 503 & 703 & 765 \\ \mathbf{y}: & 512 & 677 \end{cases}$$

Then

$$\mathbf{z}: \quad 87 \quad 503 \qquad \begin{cases} \mathbf{x}: & 703 & 765 \\ \mathbf{y}: & 512 & 677 \end{cases}$$

and

$$\mathbf{z}: \quad 87 \quad 503 \quad 512 \qquad \begin{cases} \mathbf{x}: & 703 & 765 \\ \mathbf{y}: & 677 \end{cases}$$

Next,

$$\mathbf{z}: \quad 87 \quad 503 \quad 512 \quad 677 \qquad \begin{cases} \mathbf{x}: & 703 & 765 \\ \mathbf{y}: \end{cases}$$

Since $\mathbf{y}$ is exhausted now, we simply append the rest of $\mathbf{x}$ to $\mathbf{z}$. So, finally:

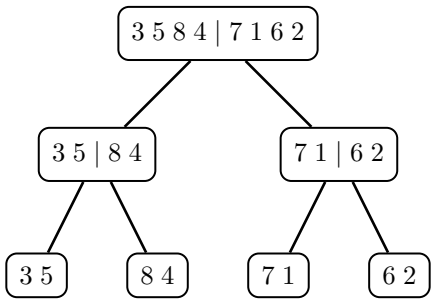
$$\mathbf{z} = [87, 503, 512, 677, 703, 765]$$

If $\mathbf{x}$ has n items and $\mathbf{y}$ m items, then the number of comparisons needed for a two-way merge is essentially proportional to $m + n$, so obviously, merging is simpler than sorting.

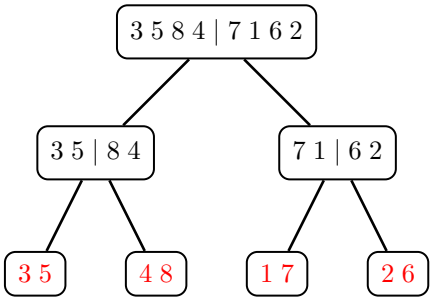
14.1.4 Divide and Conquer

But what does merging help to solve the more complicated task of sorting? The idea is the old roman principle *divide et impera*, divide and conquer. We may split a long file into two parts of more or less equal size, sort each part separately and then merge the sorted subfiles. But each subfile may again be split into two parts, sorted and then parts are merged. This splitting process can be continued until we arrive at subfiles having at most 2 items. But sorting these is trivial, at most one comparison is needed. The result will be a *binary tree structure*.

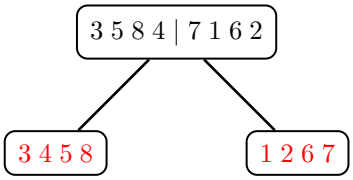
Here is an example: let $\mathbf{a} = [3, 5, 8, 4, 7, 1, 6, 2]$. Recursive splitting yields the tree structure:



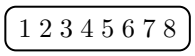
Next sort the leaves of the tree, each leave requires one comparison and one exchange:



Now merge the leaves:



One more merge yields the sorted array:



Easy, isn't it? This sorting algorithm, commonly known as *mergesort*, is one of the very first methods proposed for sorting by John von Neumann as early as 1945.

And merge sort is very effective! It can be shown (see the bonus problem below) that the number of comparisons required by *mergesort* equals:

$$C_n = n\lfloor \lg n \rfloor + 2n - 2^{\lfloor \lg n \rfloor + 1}, \tag{14.1}$$

where lg is the logarithm to base 2: $\lg x = \ln x / \ln 2$ and $\lfloor x \rfloor$ is the so-called *floor function*, i.e., round x to the next lower integer. For instance $\lfloor 5.8 \rfloor = 5$.

Formula (1) is quite remarkable in many respects:

- It is an exact formula, it is the solution of a special type of the so-called *master equation*:

$$C_n = C_{\lfloor n/2 \rfloor} + C_{\lceil n/2 \rceil} + n, \tag{14.2}$$

  here $\lceil x \rceil$ denote the *ceiling function*, i.e. round x the the next largest integer. Equation (2) and its various companions occurring in other types of divide-and-conquer algorithms like searching items in files, fast multiplication of integers, etc., have many interesting properties, see also the *bonus problem* below.

- A closer look at (1) reveals, that the number of comparison is dominated by the first term

$$n\lfloor \lg n \rfloor \leq n \lg n \doteq 1.4427 \cdot n \ln n$$

  What you can see here is indeed a very strong statement, the *Fundamental Theorem of Sorting*. It says that no comparison based sorting algorithm can guarantee to sort n items with fewer than $\lg(n!) \sim n \lg n$ comparisons. This is a theoretical lower bound, and no sorting algorithm exists which can be better, provided, (a) it is based on comparisons and (b) does not utilize additional information, e.g. about the distribution of keys (see below).

Let us compare *selection sort* and *mergesort* when applied to a file with $n = 100000$ keys. The exact number of comparisons needed by *selection sort* is

$$C_{\text{ss}} = \frac{n(n-1)}{2} = 4999950000,$$

For *mergesort* we have by (1):

$$C_{\text{ms}} = 1668928,$$

this is a reduction of the number of comparisons by 99.967 %! Thus *mergesort* outperforms *selection sort* dramatically.

The $n \lg n$ bound is the best possible and it is attained by *mergesort*. But there are also other sorting algorithms with are roughly as efficient as *mergesort*.

A serious competitor is *quicksort*. This is the most famous and most frequently used algorithm and it relies also on the idea of *divide and conquer*. The file is recursively (and very cleverly) partitioned into subfiles and these are sorted *in place*, so there is no merging process and no extra storage space is required.

14.2 Where to go from here

14.2.1 Issues of general interest

- Give an overview of comparison based sorting, in particular describe simple algorithms like selection sort, insertion sort, bubble sort and Shell sort. There are also some rather weird algorithms like *bogosort*, also known as *stupid sort*.
- Compare these algorithms and discuss their behavior when applied on almost sorted data.
- Discuss *mergesort* and *quicksort* in detail. Explain also the algorithm *heapsort*, an interesting alternative which is not based on the paradigm of divide and conquer.
- Give a derivation of the *Fundamental Theorem of Sorting*, Cormen et al. (2001, chapter 9) will be helpful, but also Sedgewick and Wayne (2011, chapter 2)
- Which sorting algorithms are used by popular programming languages like C, java, python, etc?

14.2.2 Some more suggestions

- Under certain circumstances it is possible to sort n keys in *linear time* by a number of comparisons $C_n \leq Mn$, where M is a constant independent of n. Note that $Mn \leq n \lg n$ for sufficiently large n. Describe such an algorithm. Under what conditions is such a fast sort possible?
- Give a general discussion of divide and conquer algorithms. Formulate the *master equation* and discuss the (asymptotic) properties of its solutions. In particular solve the master equation for *mergesort*, i.e., derive formula (1). Dasgupta, Papadimitriou, and Vazirani (2008, chapter 2) may be helpful, and very helpful is certainly Sedgewick and Flajolet (2013, chapter 2).

Note. Your own ideas and creativity are always welcome!

14.3 An Annotated Bibliography

The textbook by Cormen et al. (2001) has a very easy-to-read introduction the the mathematical foundations of sorting and other algorithms. Part II of this book (4 chapters) is exclusively devoted to sorting.

Dasgupta, Papadimitriou, and Vazirani (2008, chapter 2) gives a simple introduction to the idea of *divide and conquer*. A very detailed coverage of this important principle is given in the wonderful book Sedgewick and Flajolet (2013). Here you find also a careful exposition of standard mathematical techniques to solve divide and conquer recurrences and for their asymptotic analysis.

The classical textbook is certainly Knuth (1998). It contains an incredible wealth of material, though this text is not so easy to read.

14.4 References

[1] Thomas H. Cormen et al. *Introduction to Algorithms*. 2nd. McGraw-Hill Higher Education, 2001.

[2] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh Vazirani. *Algorithms*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2008. URL: http://cseweb.ucsd.edu/~dasgupta/book/toc.pdf.

[3] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998.

[4] Robert Sedgewick and Philippe Flajolet. *An Introduction to the Analysis of Algorithms*. Boston, MA, USA: Addison-Wesley, 2013.

[5] Robert Sedgewick and Kevin Wayne. *Algorithms, 4th Edition*. Addison-Wesley, 2011.

Remark. The book Knuth (1998) is *the classical text* on the subject!

Topic 15

Women in Mathematics

From Hypatia to Emmy Noether and beyond

Justifably proud, for you were a great woman mathematician - I have no reservations in calling you the greatest that history has known.

Hermann Weyl, 1935

Keywords: *history of mathematics, female mathematicians*

15.1 An Invitation

15.1.1 Headline news

On August 12, 2014 The New York Times headlined: *Top Math Prize Has Its First Female Winner*, and one day later The Guardian followed with: *Fields Medal mathematics prize won by woman for first time in its history!*

Indeed, these were exciting news. Maryam Mirzakhani, who was born and raised in Iran, has been awarded the highest honor a mathematician can attain, the Fields Medal. It is the world's most prestigious mathematics prize and for the first time since the award was established nearly 80 years ago it was awarded to a woman.

Maryam Mirzakhani
(1977–2017)

The New York Times commented on this occasion: *While women have reached parity in many academic fields, mathematics is still dominated by men, who earn about 70 percent of the doctoral degrees. The disparity is even more striking at the highest echelons. Since 2003, the Norwegian Academy of Science and Letters has awarded the Abel Prize, recognizing outstanding mathematicians with a monetary award of about $ 1 million; all 14 recipients so far are men. No woman has won the Wolf Prize in Mathematics, another prestigious award.*

15.1.2 Emmy Noether

Emmy Noether (1882-1935) is an impressing example of the problems women were facing when they pursued a career as academic mathematicians. Noether was invited by Felix Klein and David Hilbert to the University of Göttingen. But although being an outstanding and extremely prolific mathematical talent, she was not allowed to get the *venia docendi*, i.e., become a docent with the right to hold lectures at a university. Members of the faculty argued against Noether, that being a docent she will also give examinations and it may happen, that male students may fail such an exam. Incredible, a man failing an exam held by a woman! It is reported that David Hilbert replied angrily: *Aber meine Herren, eine Universität ist doch keine Badeanstalt!* As a result Emmy Noether had to announce and hold her lectures under the name of David Hilbert. In April 1933 she was expelled from Göttingen University by Nazi authorities, emigrated to the United States and got a poorly paid teaching position at Bryn Mawr College. Only two years later she died at an age of 53 years.

Emmy Noether
(1882-1935)

15.1.3 Other remarkable women

There are many other impressing examples of female mathematicians, just to mention a few: Hilda Geiringer (1893-1973), the first woman getting a doctor's degree at University of Vienna in 1917. Her biography would be perfect material for a novel or a film. By the way, Hilda Geiringer spent also some time at Bryn Mawr College. Or, Olga Taussky-Todd (1906-1995), who became a torchbearer for matrix theory. And there are many other extraordinary female mathematicians. We find them though sporadically in all epochs, from ancient times to our days. An ancient example is Hypatia of Alexandria (around 350 - 415 AD), an example from 18th century is Maria Gaetana Agnesi (1718 - 1799), and a really remarkable women is certainly Sofia (Sonya) Kovalevskaya (1850 - 1891).

The lives of these and other women were often tragic, her achievements remarkable and outstanding even more when we recognize the difficulties and social and academic opposition they had to overcome. But, fortunately, times are changing and today more and more female mathematicians are fully respected in academic and non-academic society.

15.2 Where to go from here

15.2.1 What to be avoided

Let me first tell you what *I do not want*: I will not accept an unsophisticated collage of biographical sketches composed by cut and paste from various internet resources.

15.2.2 What you should do

Although the topic of this thesis is certainly located in the field of history of mathematics, it lies at the frontier to other disciplines, in particular to sociology of science. This is an important point regarding methodology.

So, what I want is this:

- A serious discussion of the role of women in mathematics. You should work out clearly how the social perception of this question changed over time, in particular since World War II.
- Find out important facts about the social backgrounds of female mathematicians. What about their families, their parents? Who discovered their mathematical talents, who were their mentors? Find similarities and explain them.
- Reputation of a professional mathematician is usually strongly connected to academic positions. A good indicator are renown professorships held by women. For example Alice Chang is Eugene Higgins Professor of Mathematics at Princeton University since 1998. Another indicator are prizes and awards like the Fields Medal ot the Abel Prize.
- And what about a very common problem women are facing when pursuing careers (not only in mathematics): how do they manage to combine her carreer with family, with motherhood?
- Are there areas of mathematics preferred by women, like the theory of numbers, differential geometry, statistics?
- Elaborate on the New York Times-comment above about disparity between male and female mathematicians.
- Your argumentation should also be supported by empirical analysis. Thus you will have to collect *data* and analyze them.

15.2.3 A final remark on style

This topic is in a certain sense *non-mathematical* which does not mean that it is trivial from a methodological point of view. Indeed, this thesis does require a clear and elaborate methodological approach. So, suppose you are a young journalist and this is your first chance at a reknown scientific magazine. The editor-in-chief tells you: *this is your topic, write a good story about women in mathematics!*

15.3 An Annotated Bibliography

Boyer and Merzbach (2011) is the classical textbook about history of mathematics. You may fnd interesting also Osen (1975). However, this book is somewhat outdated, though still a very interesting and easy-to-read text. A wonderful paper about Emmy Noether and Hermann Weyl is Roquette (2008). It contains also Weyl's poignant funeral speech for Emmy Noether on April 18, 1935.

There are many interesting places in the internet, here are a few which I found interesting:

- *Biographies of Women Mathematicians* is a webpage maintained at Agnes Scott College, Atlanta, Georgia.
- The *IAS School of Mathematics* at Princeton University has a very good page entitled *Women in Mathematics*.
- *Smith College* in Northampton, Massachusetts, is one of the biggest and most reknown women's collages all over the world. It has a *Center for Women in Mathematics* which is part of the Department of Mathematics and Statistics.
- The *Canadian Mathematical Society* has a very well organized page *Resources for Women in Mathematics*.
- Last but not least: the *MacTutor History of Mathematics Archive* at the University of St. Andrews.

15.4 References

[1] Carl B. Boyer and Uta C. Merzbach. *A History of Mathematics*. John Wiley & Sons, 2011.

[2] Lynn M. Osen. *Women in Mathematics*. MIT Press, 1975.

[3] Peter Roquette. *Emmy Noether and Hermann Weyl*. 2008. URL: http://www.rzuser.uni-heidelberg.de/~ci3/weyl+noether.pdf.