Martin Loebl

Discrete Mathematics in Statistical Physics

Introductory Lectures

VIEWEG+
TEUBNER

Martin Loebl

Discrete Mathematics in Statistical Physics

Martin Loebl

Discrete Mathematics in Statistical Physics

Introductory Lectures

VIEWEG+
TEUBNER

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Dr. Martin Loebl
Charles University
Department of Applied Mathematics
Institut of Theoretical Computer Science
Malostranske 25
CZ-118 00 Praha
Tschechische Republik

E-Mail: loebl@kam.mff.cuni.cz

1st Edition 2010

Editorial Office: Ulrike Schmickler-Hirzebruch | Nastassja Vanselow

Vieweg+Teubner is part of the specialist publishing group Springer Science+Business Media.
www.viewegteubner.de

Cover design: KünkelLopka Medienentwicklung, Heidelberg
Printing company: STRAUSS GMBH, Mörlenbach
Printed on acid-free paper
Printed in Germany

ISBN 978-3-528-03219-7

dedicated to Zbyněk and Jaroslava, my parents

Preface

The purpose of these lecture notes is to briefly describe some of the basic concepts interlacing discrete mathematics, statistical physics and knot theory. I tried to emphasize a 'combinatorial common sense' as the main method. No attempt of completeness was made. The book should be accessible to the students of both mathematics and physics. I profited from previous books and expositions on discrete mathematics, statistical physics, knot theory and others, namely [B1], [BRJ], [BB], [J1], [KG], [LL], [MN], [MJ], [MT], [S0], [S3], [SM], [WFY], [WD], [KSV]. Most of the material contained in the book is introductory and appears without a reference to the original source. This book has been an idea of my editor Martin Aigner. I would like to thank to him for his support and help. Many other colleagues helped me with the book. Mihyun Kang, Jirka Matoušek, Iain Moffatt, Jarik Nešetřil, Dominic Welsh and Christian Krattenthaler read earlier versions, and without their extensive comments the book would probably not exist. I had enlightening discussions on several topics discussed in the book, in particular with Martin Klazar, Roman Kotecký, Ondřej Pangrác, Gregor Masbaum, Xavier Viennot and Uli Wagner. Marcos Kiwi saved the whole project by gently teaching me how to draw pictures amd Winfried Hochstaettler drew one; I am sure you will be able to detect it. Large part of the book was written during my visit, in the whole year 2006, at the School of Mathematics and the Centro Modelamiento Matematico, Universidad de Chile. I want to thank my colleagues there for wonderful hospitality, and gratefully acknowledge the support of CONICYT via project Anillo en Redes, ACT-08. But of course, the seminal ingredient in the process of making the book was the creative environment of my home department of applied mathematics and the institute of theoretical computer science at the Charles University, Prague.
Some theorems and observations in the book appear without a proof. Usually a pointer is given to a book (preferentially) or to a paper where a proof can be found. If no pointer is given, then I believe (possibly mistakenly) that it should be possible to prove the statement in an elementary and not very complicated way. The reader is encouraged to write down such proofs as exercises. The first five chapters concentrate on the introductory discrete mathematics. Chapters six and seven are devoted to the partition functions, and chapter eight is an introduction to the theory of knots. The last chapter describes two combinatorial technics which solve the 2D Ising and dimer problems.

Prague, September 2009
Martin Loebl

Contents

Chapter 1

Basic concepts

In this introductory chapter we first present some very basic mathematical formalism. Then we introduce algorithms and complexity. The chapter ends with basic tools of discrete calculations.

1.1 Sets, functions, structures

We will use symbols $\mathbb{N}, \mathbb{Z}, \mathbb{P}, \mathbb{Q}, \mathbb{R}$, and $\mathbb{C}$ to denote the sets of the natural, integer, positive integer, rational, real, and complex numbers. If not specified otherwise then i, j, k, m, n are non-negative integers. We sometimes denote by $[n]$ the set $\{1, \ldots, n\}$. We denote by $|X|$ the cardinality of a set X. A function f from a set X to a set Y is called *one-to-one* or *injective* if for $x, y \in X$ $x \neq y$ implies $f(x) \neq f(y)$, it is *surjective* or *onto* if for each $y \in Y$ there is $x \in X$ such that $f(x) = y$, and it is a *bijection* if it is both one-to-one and onto. If X, Y are finite then a bijection is also called a *permutation*. Let $|X| = n$ and $|Y| = m$. There are $n! = n(n-1)(n-2)\cdots 1$ permutations of X; $n!$ is the *factorial function*. The number of all functions from X to Y is m^n and the number of one-to-one functions from X to Y is $m(m-1)\cdots(m-n+1)$. The number of surjective functions does not have such a nice formula, it may be written with the help of the principle of inclusion and exclusion as

$$\sum_{i=0}^{m} (-1)^i \binom{m}{i} (m-i)^n. \tag{1.1}$$

The *Kronecker delta function* is defined by $\delta(x, y) = 1$ if $x = y$, and zero otherwise. If X is a set, then we denote by 2^X the set of all the subsets of X; $|2^X| = 2^n$. We further denote by $\binom{X}{k}$ the set of all subsets of X of cardinality k. We have

$$\left|\binom{X}{k}\right| = \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

The *binomial theorem* says that

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

The symbol $\binom{n}{k}$ is called the *binomial coefficient.* The *multinomial coefficient* is defined by

$$\binom{n}{k_1, \cdots, k_m} = \frac{n!}{k_1! \cdots k_m!}$$

and the *multinomial theorem* says that

$$(x_1 + \cdots + x_m)^n = \sum_{k_1+\cdots+k_m=n} \binom{n}{k_1, \cdots, k_m} x_1^{k_1} \cdots x_m^{k_m}.$$

A very good estimate of the factorial function $n!$ is given by Stirling's formula which approximates $n!$ by $(2\pi n)^{1/2}(\frac{n}{e})^n$.
If $Y \subset X$, then the *incidence vector* of Y will be denoted by $i(Y)$; $i(Y)$ is the $0,1$ vector indexed by the elements of X, where $[i(Y)]_z = 1$ if and only if $z \in Y$. We will sometimes not distinguish between a set and its incidence vector.
An *ordered pair* is usually denoted by (x, y) where x is the first element of the pair. A *binary relation* on X is any subset of $X \times X = \{(x, x'); x \in X, x' \in X\}$. Any function $X \to X$ is a binary relation on X. A *partially ordered set*, or *poset* for short, is a pair $(X, \preceq)$, where X is a set and $\preceq$ is a binary relation on X that is reflexive ($x \preceq x$), transitive ($x \preceq y$ and $y \preceq z$ imply $x \preceq z$), and (weakly) antisymmetric ($x \preceq y$ and $y \preceq x$ imply $x = y$). The binary relation $\preceq$ is itself called a *partial ordering.* A partial ordering where any pair of elements is comparable is called a *linear ordering.* An important example of a linear ordering is the *lexicographic ordering.* Let $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_m)$ be two strings of integers. We say that a is lexicographically smaller than b if a is an initial segment of b or $a_j < b_j$ for the smallest index j such that $a_j \neq b_j$.
Let $(X, \preceq)$ be a poset and $Y \subset X$. We say that Y is a *chain* if the induced ordering $(Y, \preceq)$ is linear.
The symbol $\mathbb{F}$ will denote a field; $\mathbb{F}$ will usually be equal to $\mathbb{R}$ or $\mathbb{C}$, or to the finite 2-element field $GF(2) = (\{0, 1\}, +, \times)$ with addition and multiplication modulo 2. The symbol $\mathbb{F}^d$ denotes the vector space of dimension d over $\mathbb{F}$. The elements of $\mathbb{F}^d$ are called vectors; for $x \in \mathbb{F}^d$ we write $x = (x_1, \cdots, x_d)$. We will understand vectors as both row and column vectors. The scalar product of two vectors x, y is $xy = x_1y_1 + \cdots + x_dy_d$. A set $\{x^1, \cdots, x^k\}$ of vectors of $\mathbb{F}^d$ is *linearly independent* if, whenever $\sum_{i=1}^{k} a_i x^i = 0$ and each $a_i \in \mathbb{F}$, then $a_1 = a_2 = \cdots = a_k = 0$. The *dimension* $\dim(\{x^1, \cdots, x^k\})$ of a set of vectors $\{x^1, \cdots, x^k\}$ is the maximum number of linearly independent elements in $\{x^1, \cdots, x^k\}$. A *subspace* of a vector space $\mathbb{F}^d$ is any subset of $\mathbb{F}^d$ which is closed under addition, and multiplication by an element of $\mathbb{F}$. Two subspaces X, Y are *isomorphic* if there is a bijection $f : X \to Y$ such that for each $a, b \in X$ and $c \in \mathbb{F}$, $f(a + b) = f(a) + f(b)$ and $f(ca) = cf(a)$. The *orthogonal complement*

of a subspace $X \subset \mathbb{F}^d$ is the subspace $\{y \in \mathbb{F}^d; xy = 0 \text{ for each } x \in X\}$.
Let $A = (a_{ij})$ be a matrix of n rows and m columns, with entries from field $\mathbb{F}$. We say that A is an $n \times m$ matrix. If $n = m$ then A is a *square matrix*. The *determinant* of a square $n \times n$ matrix A is defined by $\det(A) = \sum_\pi (-1)^{\text{sign}(\pi)} \prod_{i=1}^n a_{\pi(i)i}$, where the sum is over all permutations π of $1, \cdots, n$ and $\text{sign}(\pi) = |\{i < j; \pi(i) > \pi(j)\}|$. The determinant characterizes linearly independent vectors. A set of n vectors of length n is linearly independent if and only if $\det(A) \neq 0$, where A is the matrix whose set of columns (or rows) is formed by the vectors. The determinant of a matrix may be calculated efficiently by the *Gaussian elimination*. The *permanent* of matrix A is defined analogously as the determinant, but the $(-1)^{\text{sign}(\pi)}$ term is omited from each summand. Hence, $\text{Per}(A) = \sum_\pi \prod_{i=1}^n a_{\pi(i)i}$. There is no efficient algorithm to calculate the permanent. The *identity matrix* is the square matrix A such that $a_{ii} = 1$ and $a_{ij} = 0$ for $i \neq j$. The *trace* of a square matrix A, denoted by $\text{tr}(A)$, is defined by $\text{tr}(A) = \sum_i a_{ii}$.
The symbol $\mathbb{R}^d$ also denotes the Euclidean space of dimension d. A *curve* in $\mathbb{R}^d$ is the image of a continuous function $f : [0, 1] \to \mathbb{R}^d$. A curve is *simple* if it is one-to-one, and it *connects* its endpoints $f(0), f(1)$. A curve is *closed* if $f(0) = f(1)$. The *Euclidean norm* of $x \in \mathbb{R}^d$ is $||x|| = (xx)^{1/2}$. A set $\{x^0, x^1, \cdots, x^k\}$ of vectors of $\mathbb{R}^d$ is *affinely independent* if, whenever $\sum_{i=0}^k a_i x^i = 0$, $\sum_{i=0}^k a_i = 0$ and each $a_i \in \mathbb{R}$, then $a_0 = a_1 = \cdots = a_k = 0$. For two points x_0, x_1 affine independence means $x^0 \neq x^1$; for three points it means that x^0, x^1, x^2 do not lie on a common line; for four points it means that they do not lie on a common plane; and so on. The *rank* of a set of points of $\mathbb{R}^d$, denoted by $\text{rank}(\{x^0, \cdots, x^k\})$, is the maximum number of affinely independent elements in $\{x^0, \cdots, x^k\}$.
There is a simple relation between linear and affine independence: $x^0, \cdots, x^k$ are affinely independent if and only if $x^1 - x^0, \cdots, x^k - x^0$ are linearly independent. This happens if and only if the $(d+1)$-dimensional vectors $(1, x^0), \ldots, (1, x^k)$ are linearly independent. An *affine subspace* is any subset $A \subset \mathbb{R}^d$ which contains, for each pair of its elements x, y, the line through x, y. A *hyperplane* in $\mathbb{R}^d$ is a $(d-1)$-dimensional affine subspace, i.e., a set of the form $\{x \in \mathbb{R}^d : ax = b\}$ for some nonzero $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$. A (closed) *half-space* has the form $\{x \in \mathbb{R}^d : ax \leq b\}$ for some nonzero $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$.
A set $C \subset \mathbb{R}^d$ is *convex* if for every $x, y \in C$, the segment $\{ax + (1-a)y : 0 \leq a \leq 1\}$ between x and y is contained in C. The *convex hull* of a set $X \subset \mathbb{R}^d$ is the intersection of all convex sets containing X, and it is denoted by $\text{conv}(X)$. Each $x \in \text{conv}(X)$ may be written as a *convex combination* of elements of X: there are $x^1, \cdots, x^k \in X$ and real numbers $a_1, \cdots, a_k \geq 0$ such that $\sum_{i=1}^k a_i = 1$ and $x = \sum_{i=1}^k a_i x^i$.
A *convex polytope* is the convex hull of a finite subset of $\mathbb{R}^d$. Each convex polytope can be expressed as the intersection of finitely many half-spaces. Conversely, by the Minkowski-Weyl theorem, if an intersection of finitely many half-spaces is bounded, then it is a convex polytope. A *face* of a convex polytope P is P itself or a non-empty intersection of P with a hyperplane that does not dissect P (i.e., not both of the open half-spaces defined by the hyperplane

intersect P in a non-empty set).

1.2 Algorithms and Complexity

Algorithmic considerations are important for many concepts of both discrete mathematics and statistical physics. We make only basic algorithmic remarks in this book, and therefore the following exposition on algorithms and complexity is very brief.

Informally, an *algorithm* is a set of instructions to be carried out mechanically. Applying an algorithm to its *input* we get some *output*, provided that the sequence of the instructions prescribed by applying the algorithm terminates. The application of an algorithm is often called a *computation*. Usually inputs and outputs are strings (words, finite sequences) from a finite alphabet; a basic example are binary words, i.e., finite sequences of $0, 1$. The notion of an algorithm is usually formalized by the definition of a Turing machine.

A *Turing machine* consists of the following components:

- a finite set S called the *alphabet*,
- an element $b \in S$ called the *blank symbol*,
- a subset $A \subset S$ called the *external alphabet*; we assume $b \notin A$,
- a finite set Q whose elements are called *states* of the Turing machine,
- an initial state $s \in Q$,
- a *transition function*, i.e., a function

$$t : Q \times S \to Q \times S \times \{-1, 0, 1\}.$$

A Turing machine has a *tape* that is divided into cells. Each cell carries one symbol from S. We assume that the tape is infinite, thus the content of the tape is an infinite sequence $s = s_0, s_1, \cdots$ of elements of S.

A Turing machine also has a read-write *head* that moves along the tape and changes symbols. If the head is in position p along the tape, it can read symbol s_p and write another symbol in its place.

The behaviour of a Turing machine is determined by a control device. At each step of the computation, this device is in some state $q \in Q$. The state q and the symbol s_p under the head determine the action performed by the Turing machine: the value of the transition function, $t(q, s_p) = (q', s', p')$, contains the new state q', the new symbol s' to be written in the place of s_p, and the shift $p' \in \{-1, 0, 1\}$ of the position of the head. If the head bumps into the left boundary of the tape (that happens when $p + p' < 0$), then the computation *stops*.

Next we describe the input given to the Turing machine, and how the output is obtained. Let A^* denote the set of all the strings (finite sequences) of elements

of A. Inputs and outputs to the Turing machine with the external alphabet A are strings from A^*. An input string I is written on the tape and followed by the blank symbol b. Initially, the head is at the beginning (left end) of I. If the Turing machine stops (by bumping into the left boundary of the tape), we read the tape from left to right starting from the left end until we reach some symbol that does not belong to A. The initial segment of the tape until that symbol will be the output of the Turing machine.
Every Turing machine *computes* a function from a subset of A^* to A^*. There are functions that are *not computable*. A Turing machine is obviously an algorithm in the informal sense. The converse assertion is called the

Church-Turing thesis: Any algorithm can be realised by a Turing machine. Note that the Church-Turing thesis is not a mathematical theorem, but rather a statement about our understanding of the informal notion of algorithm.

Complexity classes. The computability of a function does not guarantee that we can compute it in practice since an algorithm may require too much time. The idea of an effective algorithm is usually formalized by the notion of *polynomial algorithms*. We say that a function T on the positive integers is of *polynomial growth* if $T(n) \leq cn^d$ for all n and some constants c, d. We say that a function f defined on the binary strings of $\{0,1\}^*$ is *computable in polynomial time* if there exists a Turing machine that computes f in time $T(n)$ of polynomial growth, where n is the length of the input. Such a Turing machine is called a *polynomial algorithm*. Polynomial time *encoding* plays a crucial role. For instance, if the input is an integer N in the *unary* representation then the input size is $|N|$ but if the representation is binary, the input size is only $\log(|N|)$. The class of all functions computable in polynomial time is denoted by P. We should remark here that computability in polynomial time does not guarantee practical computability either, but it is a good indication for it.
A special class of algorithmic problems are the *decision problems*. In a decision problem, we want the answer to be *yes* or *no*. This clearly may be modeled as a function from a subset of A^* to $\{0,1\}$ where 0 encodes *no* and 1 encodes *yes*. It is customary to call such functions *predicates*. One can think about predicates as about properties: the predicate indicates for each string whether it has the property (yes) or does not have the property (not). Hence the algorithmic problem to compute a predicate may be formulated as the algorithmic problem to test the corresponding property.
Another basic complexity class, the class NP, is usually defined only for the predicates. We say that a predicate $R(x,y)$, where x and y are binary strings, is *polynomially decidable* if there is a Turing machine that computes it in time of polynomial growth (the size of the input is $|x| + |y|$).
The class NP is the class of all predicates f for which there is a polynomial growth function $T(n)$ and a polynomially decidable predicate R of two variables so that $f(x) = 1$ if and only if there is y such that $|y| < T(|x|)$ and $R(x,y) = 1$. Informally, NP is the class of the predicates (i.e., properties), for which there is a certificate (coded by y) that can be checked in polynomial time. Most of the properties discussed in this book belong to NP.
Clearly $P \subset NP$. Over the past 30 years intensive research has been directed

towards proving that the inclusion is strict. The question whether $P \neq NP$ is today one of the fundamental problems of both mathematics and computer science.

Reducibility. When can we say that one problem is algorithmically at least as hard as another problem? We model the efficiency by the polynomial time complexity, and so the answer is naturally given by the following notion of *polynomial reducibility*: we say that a predicate f_1 is *polynomially reducible* to a predicate f_2 if there exists a function $g \in P$ so that $f_1(x) = f_2(g(x))$, for each input string x.

A predicate $f \in NP$ is called NP*-complete* if any predicate in NP is polynomially reducible to it. The predicates that are NP-complete are the most difficult predicates of NP: if some NP-complete predicate is in P then $P = NP$. It is customary to speak about NP*-complete problems* rather than NP-complete predicates. The existence of an efficient algorithm to solve an NP-complete problem is considered to be very unlikely.

A seminal result in algorithmic complexity is that NP-complete predicates (problems) exist. This was proved independently by Cook and Levin. Many natural NP-complete problems are known, see [GJ].

1.3 Generating functions

A useful way of counting is provided by generating functions. If f is a function from the non-negative integers, we can consider its (ordinary) *generating function* $\sum_{n\geq 0} f(n)x^n$ and its *exponential generating function* $\sum_{n\geq 0} f(n)x^n/n!$.

The generating functions are *formal power series*, since we are not concerned with letting x take particular values, and we ignore questions of convergence. This formalism is convenient since we can perform various operations on the formal power series, for instance

$$\left(\sum_{n\geq 0} a_n x^n\right) + \left(\sum_{n\geq 0} b_n x^n\right) = \sum_{n\geq 0} (a_n + b_n)x^n,$$

$$\left(\sum_{n\geq 0} a_n x^n/n!\right) + \left(\sum_{n\geq 0} b_n x^n/n!\right) = \sum_{n\geq 0} (a_n + b_n)x^n/n!$$

and

$$\left(\sum_{n\geq 0} a_n x^n\right)\left(\sum_{n\geq 0} b_n x^n\right) = \sum_{n\geq 0} c_n x^n$$

$$\left(\sum_{n\geq 0} a_n x^n/n!\right)\left(\sum_{n\geq 0} b_n x^n/n!\right) = \sum_{n\geq 0} d_n x^n/n!$$

where $c_n = \sum_{i=0}^{n} a_i b_{n-i}$ and $d_n = \sum_{i=0}^{n} \binom{n}{i} a_i b_{n-i}$.

These operations coincide with the addition and multiplication of functions when the power series converge for some values of x. Let us denote by $\mathbb{C}[[x]]$ the set of all formal power series $\sum_{n\geq 0} a_n x^n$ with complex coefficients. Addition and multiplication in $\mathbb{C}[[x]]$ are clearly commutative, associative and distributive, thus $\mathbb{C}[[x]]$ forms a commutative ring where 1 is the unity. Formal power series with the coefficients in a non-commutative ring (like the square matrices of the same size) are also extensively considered; they form a non-commutative ring with unity.

If $F(x)$ and $G(x)$ are elements of $\mathbb{C}[[x]]$ satisfying $F(x)G(x) = 1$ then we write $G(x) = F(x)^{-1}$. It is easy to see that $F(x)^{-1}$ exists if and only if $a_0 = F(0) \neq 0$. If $F(x)^{-1}$ exists then it is uniquely determined. We have $((F(x)^{-1})^{-1} = F(x)$.

Example 1.3.1. Let $a \neq 0$ and $(\sum_{n\geq 0} a^n x^n)(1 - ax) = \sum_{n\geq 0} c_n x^n$, where a is a non-zero complex number. Then from the definition of multiplication we get $c_0 = 1$ and $c_n = 0$ for $n > 0$. Hence we may write

$$\sum_{n\geq 0} a^n x^n = (1 - ax)^{-1}.$$

The identity may be derived in the same way in every ring of formal power series over a (not necessarily commutative) ring with unity. Hence, for instance, for square complex matrices it can be written as

$$\sum_{n\geq 0} A^n x^n = (I - Ax)^{-1}.$$

This is of course just the formula for summing a geometric series. Informally speaking, if we have an identity involving power series that is valid when the power series are regarded as functions (when the variables are sufficiently small complex numbers), then the identity remains valid when regarded as an identity among formal power series. Formal power series may naturally have more than one variable.

1.4 Principle of inclusion and exclusion

Let us start with the introduction of a paper of Whitney, which appeared in Annals of Mathematics in August 1932:

"Suppose we have a finite set of objects (for instance books on a table), each of which either has or has not a certain given property A (say of being red). Let n be the total number of objects, $n(A)$ the number with the property A, and $n(\bar{A})$ the number without the property A. Then obviously $n(\bar{A}) = n - n(A)$. Similarly, if $n(AB)$ denotes the number with both properties A and B, and $n(\bar{A}\bar{B})$ the number with neither property, then $n(\bar{A}\bar{B}) = n - n(A) - n(B) + n(AB)$, which is easily seen to be true. The extension of these formulas to the general case where any number of properties is considered is quite simple, and is well known to logicians. It should be better known to mathematicians also; we give in this paper several applications which show its usefulness."

It is known today, under the name *principle of inclusion and exclusion* (PIE).