Preface

Combinatorial mathematics has been pursued since time immemorial, and at a reasonable scientific level at least since Leonhard Euler (1707–1783). It rendered many services to both pure and applied mathematics. Then along came the prince of computer science with its many mathematical problems and needs – and it was combinatorics that best fitted the glass slipper held out. Moreover, it has been gradually more and more realized that combinatorics has all sorts of deep connections with "mainstream areas" of mathematics, such as algebra, geometry and probability. This is why combinatorics is now a part of the standard mathematics and computer science curriculum.

This book is as an introduction to *extremal combinatorics* – a field of combinatorial mathematics which has undergone a period of spectacular growth in recent decades. The word "extremal" comes from the nature of problems this field deals with: if a collection of finite objects (numbers, graphs, vectors, sets, etc.) satisfies certain restrictions, how large or how small can it be?

For example, how many people can we invite to a party where among each three people there are two who know each other and two who don't know each other? An easy Ramsey-type argument shows that at most five persons can attend such a party. Or, suppose we are given a finite set of nonzero integers, and are asked to mark an as large as possible subset of them under the restriction that the sum of any two marked integers cannot be marked. It appears that (independent of what the given integers actually are!) we can always mark at least one-third of them.

Besides classical tools, like the pigeonhole principle, the inclusion-exclusion principle, the double counting argument, induction, Ramsey argument, etc., some recent weapons – the probabilistic method and the linear algebra method – have shown their surprising power in solving such problems. With a mere knowledge of the concepts of linear independence and discrete probability, completely unexpected connections can be made between algebra, probability, and combinatorics. These techniques have also found striking applications in other areas of discrete mathematics and, in particular, in the theory of computing.

Nowadays we have comprehensive monographs covering different parts of extremal combinatorics. These books provide an invaluable source for students and researchers in combinatorics. Still, I feel that, despite its great po-

tential and surprising applications, this fascinating field is not so well known for students and researchers in computer science. One reason could be that, being comprehensive and in-depth, these monographs are somewhat too difficult to start with for the beginner. I have therefore tried to write a "guide tour" to this field – an introductory text which should

- be self-contained,
- be more or less up-to-date,
- present a wide spectrum of basic ideas of extremal combinatorics,
- show how these ideas work in the theory of computing, and
- be accessible for graduate and motivated undergraduate students in mathematics and computer science.

Even if not all of these goals were achieved, I hope that the book will at least give a first impression about the power of extremal combinatorics, the type of problems this field deals with, and what its methods could be good for. This should help students in computer science to become more familiar with combinatorial reasoning and so be encouraged to open one of these monographs for more advanced study.

Intended for use as an introductory course, the text is, therefore, far from being all-inclusive. Emphasis has been given to theorems with elegant and beautiful proofs: those which may be called the gems of the theory and may be relatively easy to grasp by non-specialists. Some of the selected arguments are possible candidates for *The Book*, in which, according to Paul Erdős, God collects the perfect mathematical proofs. [1] I hope that the reader will enjoy them despite the imperfections of the presentation.

Extremal combinatorics itself is much broader. To keep the introductory character of the text and to minimize the overlap with existing books, some important and subtle ideas (like the shifting method in extremal set theory, applications of Janson's and Talagrand's inequalities in probabilistic existence proofs, use of tensor product methods, etc.) are not even mentioned here. In particular, only a few results from extremal graph theory are discussed and the presentation of the whole Ramsey theory is reduced to the proof of one of its core results — the Hales–Jewett theorem and some of its consequences. Fortunately, most of these advanced techniques have an excellent treatment in existing monographs by Bollobás (1978) on extremal graph theory, by Babai and Frankl (1992) on the linear algebra method, by Alon and Spencer (1992) on the probabilistic method, and by Graham, Rothschild and Spencer (1990) on Ramsey theory. We can therefore pay more attention to the recent *applications* of combinatorial techniques in the theory of computing.

A possible feature and main departure from traditional books in combinatorics is the choice of topics and results, influenced by the author's twenty

[1] "You don't have to believe in God but, as a mathematician, you should believe in The Book." (Paul Erdős)

For the first approximation see M. Aigner and G.M. Ziegler, *Proofs from THE BOOK*. Second Edition, Springer, 2000.

years of research experience in the theory of computing. Another departure is the inclusion of combinatorial results that originally appeared in computer science literature. To some extent, this feature may also be interesting for students and researchers in combinatorics. The corresponding chapters and sections are: 2.3, 4.8, 6.2.2, 7.2.2, 7.3, 10.4–10.6, 12.3, 14.2.3, 14.5, 15.2.2, 16, 18.6, 19.2, 20.5–20.9, 22.2, 24, 25, 26.1.3, and 29.3. In particular, some recent applications of combinatorial methods in the theory of computing (a new proof of Haken's exponential lower bound for the resolution refutation proofs, a non-probabilistic proof of the switching lemma, a new lower bounds argument for monotone circuits, a rank argument for boolean formulae, lower and upper bounds for span programs, highest lower bounds on the multi-party communication complexity, a probabilistic construction of surprisingly small boolean formulas, etc.) are discussed in detail.

Teaching. The text is *self-contained*. It assumes a certain mathematical maturity but *no* special knowledge in combinatorics, linear algebra, probability theory, or in the theory of computing — a standard mathematical background at undergraduate level should be enough to enjoy the proofs. All necessary concepts are introduced and, with very few exceptions, all results are proved before they are used, even if they are indeed "well-known." Fortunately, the problems and results of combinatorics are usually quite easy to state and explain, even for the layman. Its accessibility is one of its many appealing aspects.

The book contains much more material than is necessary for getting acquainted with the field. I have split it into 29 relatively short chapters, each devoted to a particular proof technique. I have tried to make the chapters almost *independent*, so that the reader can choose his/her own order to follow the book. The (linear) order, in which the chapters appear, is just an extension of a (partial) order, "core facts first, applications and recent developments later." Combinatorics is broad rather than deep, it appears in different (often unrelated) corners of mathematics and computer science, and it is about techniques rather than results – this is where the independence of chapters comes from.

Each chapter starts with results demonstrating the particular technique in the simplest (or most illustrative) way. The relative importance of the topics discussed in separate chapters is not reflected in their length – only the topics which appear for the first time in the book are dealt with in greater detail. To facilitate the understanding of the material, over 300 exercises of varying difficulty, together with hints to their solution, are included. This is a vital part of the book – many of the examples were chosen to complement the main narrative of the text. I have made an attempt to grade them: problems marked by "−" are particularly easy, while the ones marked by "+" are more difficult than unmarked problems. The mark "(!)" indicates that the exercise may be particularly valuable, instructive, or entertaining. Needless to say, this grading is subjective. Some of the hints are quite detailed so that they

actually sketch the entire solution; in these cases the reader should try to fill out all missing details.

Feedback to the author. I have tried to eliminate errors, but surely some remain. I hope to receive mail offering suggestions, praise, and criticism, comments on attributions of results, suggestions for exercises, or notes on typographical errors. I am going to maintain a website that will contain a (short, I hope) list of errata, solutions to exercises, feedback from the readers, and any other material to assist instructors and students. The link to this site as well as my email address can be obtained from the Springer website

http://www.springer.de/comp/

Please send your comments either to my email address or to my permanent address: Institute of Mathematics, Akademijos 4, 2600 Vilnius, Lithuania.

Acknowledgments. I would like to thank everybody who was directly or indirectly involved in the process of writing this book. First of all, I am grateful to Alessandra Capretti, Anna Gál, Thomas Hofmeister, Daniel Kral, G. Murali Krishnan, Martin Mundhenk, Gurumurthi V. Ramanan, Martin Sauerhoff and P.R. Subramania for comments and corrections.

Although not always directly reflected in the text, numerous earlier discussions with Anna Gál, Pavel Pudlák, and Sasha Razborov on various combinatorial problems in computational complexity, as well as short communications with Noga Alon, Aart Blokhuis, Armin Haken, Johan Håstad, Zoltan Füredi, Hanno Lefmann, Ran Raz, Mike Sipser, Mario Szegedy, and Avi Wigderson, have broadened my understanding of things. I especially benefited from the comments of Aleksandar Pekec and Jaikumar Radhakrishnan after they tested parts of the draft version in their courses in the BRICS International Ph.D. school (University of Aarhus, Denmark) and Tata Institute (Bombay, India), and from valuable comments of László Babai on the part devoted to the linear algebra method.

I would like to thank the Alexander von Humboldt Foundation and the German Research Foundation (Deutsche Forschungsgemeinschaft) for supporting my research in Germany since 1992. Last but not least, I would like to acknowledge the hospitality of the University of Dortmund, the University of Trier and the University of Frankfurt; many thanks, in particular, to Ingo Wegener, Christoph Meinel and Georg Schnitger, respectively, for their help during my stay in Germany. This was the time when the idea of this book was born and realized. I am indebted to Hans Wössner and Ingeborg Mayer of Springer-Verlag for their editorial help, comments and suggestions which essentially contributed to the quality of the presentation in the book.

My deepest thanks to my wife, Daiva, and my daughter, Indrė, for being there.

Frankfurt/Vilnius, March 2001 Stasys Jukna

Contents

Part V. Fragments of Ramsey Theory

Prolog: What This Book Is About

Many combinatorial problems have the following "extremal" formulation. Given a finite n-element set of points, the goal is to find the maximum (or minimum) possible cardinality of a system of its subsets satisfying certain assumptions. To get a feeling about what kind of problems this book deals with, we list several typical examples. (Although long, the list is far from being exhaustive.) The number(s) in brackets indicate the section(s), where the corresponding problem is discussed.

Graphs: acquaintances and strangers

- In a town with n inhabitants, how many acquaintances can there be if we know that among any k inhabitants at least two of them are strangers? For $k = 3$ the answer "at most $n^2/4$ acquaintances" was found by Mantel in 1907. Note that this is only about a half of all $n(n-1)/2$ possible acquaintances. For an arbitrary k the answer was found by Turán in 1941, and this fundamental result initiated the field, currently known as the *extremal graph theory*. [4.3, 4.4]
- We want to avoid the situation that some k of inhabitants are either mutually acquainted or are mutual strangers. Ramsey's theorem says that in any town with at least 4^k inhabitants this bad situation will definitely occur. On the other hand, using the probabilistic argument, Erdős has proved that in every town with up to $2^{k/2}$ inhabitants, there *exists* an arrangement of mutual acquaintances and strangers such that this bad situation will not appear. Using the linear algebra method, Frankl and Wilson were able even to *construct* such an arrangement if the town has up to about $k^{\log k}$ inhabitants. [27.2, 18.1, 14.3.3]

Set systems: clubs

- A town has n inhabitants and some number of clubs; each inhabitant may be a member of several (or none) of them. If no club contains all the members of another club, then we can have at most $\binom{n}{\lfloor n/2 \rfloor}$ clubs in the town. This is the classical *Sperner's theorem*. [9.2.1]

- We have m clubs $A_1, \ldots, A_m$ with s members in each, and want to know their number m. We can try to form m new "fictive" clubs $B_1, \ldots, B_m$, each with r members such that A_i and B_j will share a member if and only if $i \neq j$. If we succeed in doing so, then we know the answer: $m \leqslant \binom{s+r}{s}$. This result, due to Bollobás, generalizes Sperner's theorem and is one of the corner-stones in *extremal set theory*. [9.2.2]
- A collection of clubs forms a "sunflower" if each inhabitant, participating in at least one of them, is either a member of all or of precisely one of these clubs. A classical result of Erdős and Rado says that if each club has s members and we have more than $s!(k-1)^s$ clubs in a town, then some k of them will form a sunflower. [7.1]
- We want to form as few clubs as possible with the property that if we take any set of k inhabitants and arbitrarily split them in two groups, then there will be a club which contains among its members all the inhabitants from the first group and none from the other. It is clear that 2^n clubs are enough and that we need at least 2^k clubs (or more, if $k < n$). Using the *probabilistic method* it can be shown that, somewhat surprisingly, it is possible to achieve this situation with only about $k2^k \log n$ clubs. Such collections of clubs are important in many applications, such as testing logical circuits, construction of k-wise independent random variables, etc. [11.3, 14.4.1, 18.5]
- Each of n inhabitants participates in the lottery, where he/she can win with equal probability some amount x of points, $0 \leqslant x \leqslant N$. After that, each club calculates the total sum of points gained by its members. What is the probability that precisely one club will have the smallest (or the largest) total yield? The *isolation lemma*, due to K. Mulmuley, U. Vazirani, and V. Vazirani, says that (independent of how the clubs are formed) this will happen with probability at least $1 - n/N$. [12.3]
- The city council selects some s numbers and passes a rule that if a pair of clubs share ℓ members, then this ℓ must be among the given s numbers. How many clubs can be formed under this rule? Using the *linear algebra method* it can be proved that (no matter what the selected numbers are) the inhabitants can form at most $\sum_{i=0}^{s} \binom{n}{i}$ clubs. This far reaching extension of Fisher's inequality is the celebrated Ray-Chaudhuri–Frankl–Wilson theorem. [14.3.2]

Numbers

- A set of integers is sum-free if the sum of every two (not necessarily distinct) of its elements does not belong to it. In 1965 Erdős, using a *probabilistic argument*, proved that *every* set of N nonzero integers always contains a sum-free subset of at least size $N/3$. [20.2]
- Given an integer k, how long must a sequence of integers $a_1, \ldots, a_n$ be in order to be sure that it contains a subsequence of (not necessarily consec-

utive) elements whose sum is divisible by k? The sequence $0 \ldots 01 \ldots 1$ of $k-1$ subsequent 0's and 1's shows that the sequence must have at least $2k-1$ numbers. Using an *algebraic argument*, it can be shown that every sequence of $2k-1$ numbers will already have the desired subsequence. [28.2]

- If somebody gives us a sequence of more than sr integers, then we (without looking at that sequence) can be sure that it contains either a subsequence of s (not necessarily consecutive) increasing numbers or a subsequence of r decreasing numbers. This result was first proved by Erdős and Szekeres in 1935. In 1959 Seidenberg found a very short proof using the *pigeonhole principle*. [4.2]

Geometry

- What is the maximal set of points in the n-dimensional Euclidean space $\mathbb{R}^n$, such that all angles determined by three points from the set are strictly less than $\pi/2$? It was an old conjecture of Danzer and Grünbaum that any such set can have at most $2n-1$ points. Using the *probabilistic method*, Erdős and Füredi disproved this conjecture: there is such a set with about 1.15^n points. [21.4]
- In 1944 Hadwiger proposed the following question: how many colors do we need in order to color the points of the n-dimensional Euclidean space $\mathbb{R}^n$ so that each monochromatic set of points misses some distance? A set is said to "miss distance" d if no two of its points are at distance d apart from each other. This turns out to be quite a hard problem; the exact answer is not known even for the plane (where $n = 2$). In 1972 Larman and Rogers proved that about 2.8^n colors are enough. Using the *linear algebra method*, in 1981 Frankl and Wilson were able to prove that this exponential bound is not far from the truth: at least 1.2^n colors are necessary. [14]

Complexity theory

- Let f be a boolean function and a be an input vector. A certificate of a is a set of its bits such that looking at only these bits of a we can determine the value $f(a)$. A decision tree for f is a deterministic algorithm which, given an input a, tests its bits one-by-one in a prescribed order and outputs the value $f(a)$. Suppose we know that all inputs have certificates of size at most k. How many tests must a decision tree make on the worst case input? It turns out that k^2 tests are always enough. [10.4]
- Given a set of m 0-1 vectors, how many of their bits must be exposed in order to distinguish every single vector from the remaining vectors in the set? It turns our that, on average, it is enough to expose at most $\sqrt{m}$ bits, and there are sets for which this bound cannot be improved. [12.2]

- With every boolean function f on $2n$ variables we can associate a graph G_f whose vertices are 0-1 vectors of length n, and two vertices a, b are joined by an edge precisely when $f(a, b) = 1$. If the graph G_f has a "complicated" structure, then (intuitively) the function f should be hard to compute, that is, should require a large formula or circuit. Using the *probabilistic argument*, Razborov has proved that this intuition may be false! [24.3]
- Given a boolean function, how many And, Or and Not operations do we need to represent it as a formula? The difficulty in proving that a given boolean function has high complexity (i.e., requires large formulas, or large circuits, etc.) seems to lie in the nature of the adversary: the algorithm. Fast algorithms may work in a counterintuitive fashion, using deep, devious, and fiendishly clever ideas. How could one prove that there is no clever way to quickly solve a given problem? This has been the main issue confronting the complexity theorists since the early 1950's. We will show how, using non-trivial combinatorial arguments, this task can be solved for different models of computation – like DeMorgan formulas, combinational circuits, and span programs – under additional restrictions on the use of Not gates. [10.6, 15.2.2, 16.4, 16.5]

In this book we will learn some of the most powerful combinatorial tools which have proved useful in attacking such and similar problems:

1. basic methods: the double counting argument, the pigeonhole principle, the inclusion-exclusion formula, the averaging argument, etc.
2. the linear algebra method
3. the probabilistic method
4. Ramsey arguments.

These tools are presented in a form acceptable also to a reader from other fields of mathematics and computer science. (However, the reader should not be immediately disappointed if some of the seemingly "simple" proofs would require a half an hour of thinking – bright brains have spent maybe months to produce them!) The emphasis is made on learning methods rather than the results themselves – these are chosen to illustrate the way of reasoning in elegant and simple form.

Most of the results and techniques presented in this book are motivated by applications in the theory of computing. A fundamental problem of this theory – known as the *lower bounds problem* – is to prove that a given function cannot be computed within a given amount of resourses (time, space, chip-area, etc.). This is an extremal problem per se and we will demonstrate the role of combinatorial reasoning in its solution for different models of computation: resolution refutation proofs, boolean formulas, circuits, span programs and multi-party communication protocols (Sects. 10.4, 10.5, 10.6, 15.2.2, 16, 24, 29.3).

Notation

In this section we give the notation that shall be standard throughout the book.

Sets

We deal exclusively with finite objects. We use the standard set-theoretical notation:

$|X|$ denotes the *size* (the *cardinality*) of a set X.
A *k-set* or *k-element set* is a set of k elements.
$[n] = \{1, 2, \ldots, n\}$ is often used as a "standard" n-element set.
$A - B = \{x : x \in A \text{ and } x \notin B\}$.
$\overline{A} = X \backslash A$ is the complement of A.
$A \oplus B = (A - B) \cup (B - A)$ (symmetric difference).
$A \times B = \{(a, b) : a \in A, b \in B\}$ (Cartesian product).
$A \subseteq B$ if B contains all the elements of A.
$A \subset B$ if $A \subseteq B$ and $A \neq B$.
2^X is the set of all subsets of the set X. If $|X| = n$ then $\left|2^X\right| = 2^n$.
A *permutation* of X is a one-to-one mapping (a bijection) $f: X \to X$.
$\{0,1\}^n = \{(v_1, \ldots, v_n) : v_i \in \{0,1\}\}$ is the (binary) n-cube.
0-1 vector (matrix) is a vector (matrix) with entries 0 and 1.
An $m \times n$ matrix is a matrix with m rows and n columns.
The *incidence vector* of a set $A \subseteq \{x_1, \ldots, x_n\}$ is a 0-1 vector $v = (v_1, \ldots, v_n)$, where $v_i = 1$ if $x_i \in A$, and $v_i = 0$ if $x_i \notin A$.
The *characteristic function* of a subset $A \subseteq X$ is the function $f : X \to \{0, 1\}$ such that $f(x) = 1$ if and only if $x \in A$.

Arithmetic

Some of the results are asymptotic, and we use the standard asymptotic notation: for two functions f and g, we write $f = O(g)$ if $f \leqslant c_1 g + c_2$ for all possible values of the two functions, where c_1, c_2 are absolute constants. We write $f = \Omega(g)$ if $g = O(f)$, and $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$. If the limit of the ratio f/g tends to 0 as the variables of the functions tend to infinity, we write $f = o(g)$. Finally, $f \lesssim g$ means that $f \leqslant (1 + o(1))g$,

and $f \sim g$ denotes that $f = (1 + o(1))g$, i.e., that f/g tends to 1 when the variables tend to infinity. If x is a real number, then $\lceil x \rceil$ denotes the smallest integer not less than x, and $\lfloor x \rfloor$ denotes the greatest integer not exceeding x. As customary, $\mathbb{Z}$ denotes the set of integers, $\mathbb{R}$ the set of reals, $\mathbb{Z}_n$ an additive group of integers modulo n, and $\mathrm{GF}(q)$ (or $\mathbb{F}_q$) a finite Galois field with q elements. Such a field exists as long as q is a prime power. If $q = p$ is a prime then $\mathbb{F}_p$ can be viewed as the set $\{0, 1, \ldots, p-1\}$ with addition and multiplication performed modulo p. The sum in $\mathbb{F}_2$ is often denoted by $\oplus$, that is, $x \oplus y$ stands for $x + y \pmod 2$. We will often use the so-called *Cauchy–Schwarz inequality* (see Proposition 14.4 for a proof): if $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are real numbers then

$$\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leqslant \left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right).$$

If not stated otherwise, e = 2.718... will always denote the base of the natural logarithm.

Graphs

A *graph* is a pair $G = (V, E)$ consisting of a set V, whose members are called *vertices* (or *nodes*), and a family E of 2-element subsets of V, whose members are called *edges*. A vertex v is *incident* with an edge e if $v \in e$. The two vertices incident with an edge are its *endvertices* or *endpoints*, and the edge *joins* its ends. Two vertices u, v of G are *adjacent*, or *neighbors*, if $\{u, v\}$ is an edge of G. The number $d(u)$ of neighbors of a vertex u is its *degree*. A *walk* of length k in G is a sequence $v_0, e_1, v_1 \ldots, e_k, v_k$ of vertices and edges such that $e_i = \{v_{i-1}, v_i\}$. A walk without repeated vertices is a *path*. A walk without repeated edges is a *trail*. A *cycle* of length k is a path $v_0, \ldots, v_k$ with $v_0 = v_k$. A (connected) *component* in a graph is a set of its vertices such that there is a path between any two of them. A graph is *connected* if it consists of one component. A *tree* is a connected graph without cycles. A *subgraph* is obtained by deleting edges and vertices. A *spanning subgraph* is obtained by deleting edges only. An *induced subgraph* is obtained by deleting vertices (together with all the edges incident to them).

A *complete graph* or *clique* is a graph in which every pair is adjacent. An *independent set* in a graph is a set of vertices with no edges between them. The greatest integer r such that G contains an independent set of size r is the *independence number* of G, and is denoted by $\alpha(G)$. A graph is *bipartite* if its vertex set can be partitioned into two independent sets.

A *legal coloring* of $G = (V, E)$ is an assignment of colors to each vertex so that adjacent vertices receive different colors. In other words, this is a partition of the vertex set V into independent sets. The minimum number of colors required for that is the *chromatic number* $\chi(G)$ of G.

Set systems

A *set system* or *family of sets* $\mathcal{F}$ is a collection of sets. Because of their intimate conceptual relation to graphs, a set system is often called a *hypergraph*. A family is *k-uniform* if all its members are k-element sets. Thus, graphs are k-uniform families with $k = 2$. The *rank* of a family is the maximum cardinality of its member. A *blocking set* (or transversal) of $\mathcal{F}$ is a set which intersects every member of $\mathcal{F}$. The minimum number of elements in a blocking set of a family $\mathcal{F}$ is its *blocking number*, and is denoted by $\tau(\mathcal{F})$.

The notions of independent set and chromatic number extend to set systems. For a set system $\mathcal{F}$ over the universe X, the subset $S \subseteq X$ is called *independent* if S does not contain any member of $\mathcal{F}$. An *r-coloring* of $\mathcal{F}$ is a map $h\colon X \to \{1, 2, \ldots\}$ which assigns to each point $x \in X$ its "color" $h(x)$. Such a coloring is *legal* if none of the members of $\mathcal{F}$ is monochromatic, i.e., if for all $A \in \mathcal{F}$ there exist $x, y \in A$ such that $h(x) \neq h(y)$. The independence number $\alpha(\mathcal{F})$ and the chromatic number $\chi(\mathcal{F})$ are defined as for graphs.

Three representations

In order to prove something about families of sets (as well as to interpret the results) it is often useful to keep in mind that any family can be looked at either as a 0-1 matrix or as a bipartite graph.

Let $\mathcal{F} = \{A_1, \ldots, A_m\}$ be a family of subsets of a set $X = \{x_1, \ldots, x_n\}$. The *incidence matrix* of $\mathcal{F}$ is an $n \times m$ 0-1 matrix $M = (m_{i,j})$ such that $m_{i,j} = 1$ if and only if $x_i \in A_j$. Hence, the jth column of M is the incidence vector of the set A_j. The *incidence graph* of $\mathcal{F}$ is a bipartite graph with parts X and $\mathcal{F}$, where x_i and A_j are joined by an edge if and only if $x_i \in A_j$.

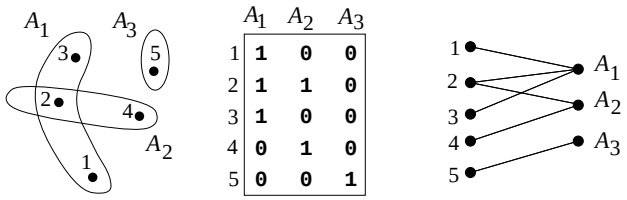

Fig. 0.1. Three representations of the family $\mathcal{F} = \{A_1, A_2, A_3\}$ over the set of points $X = \{1, 2, 3, 4, 5\}$ with $A_1 = \{1, 2, 3\}$, $A_2 = \{2, 4\}$ and $A_3 = \{5\}$

Projective planes

To justify the optimality of some results, we will often refer to the following regular families. Let $n = q^2 + q + 1$. A *projective plane* of order q is a family

of n subsets (called *lines*) of an n-element set X (of *points*) satisfying the following four conditions:

- every line has exactly $q+1$ points;
- every point belongs to exactly $q+1$ lines;
- every two lines meet in exactly one point;
- any two points lie on a unique line.

Such a family exists for any prime power q; see Chap. 13 for more details.

Boolean functions

A *boolean function* $f = f(x_1, \ldots, x_n)$ on the n variables $x_1, \ldots, x_n$ is simply a function $f\colon \{0,1\}^n \to \{0,1\}$. In particular,

$$0, \quad 1, \quad x_1 \wedge \cdots \wedge x_n, \quad x_1 \vee \cdots \vee x_n, \quad x_1 \oplus \cdots \oplus x_n$$

denote, as usual, the two constant functions, the *And* function (whose value is 1 iff $x_i = 1$ for all i), the *Or* function (whose value is 0 iff $x_i = 0$ for all i), and the *Parity* function (whose value is 0 iff $x_i = 1$ for an even number of variables x_i). For a function f, we let $\overline{f} = f \oplus 1$ denote its complement, *Not* f. The functions x_i and $\overline{x}_i$ are called *literals* (or *atoms*).

A *monomial* is an And of literals, and a *clause* is an Or of literals. The number of literals in a clause or monomial is its *length* (or *size*). The Or of an arbitrary number of monomials is a *disjunctive normal form* (*DNF*). Dually, the And of an arbitrary number of clauses is a *conjunctive normal form* (*CNF*). A boolean function f is a *t-And-Or* function if it can be written as an And of an arbitrary number of clauses, each being an Or of at most t literals. That is, a function is t-And-Or function if it can be represented by a CNF, all whose clauses have length at most t. Dually, a boolean function f an *s-Or-And* if it can be written as an Or of an arbitrary number of monomials, each being an And of at most s literals.