Reinhard Diestel

Graph Theory

Electronic Edition 2000

This is an electronic version of the second (2000) edition of the above Springer book, from their series *Graduate Texts in Mathematics*, vol. 173. The cross-references in the text and in the margins are active links: click on them to be taken to the appropriate page.

The printed edition of this book can be ordered from your bookseller, or electronically from Springer through the Web sites referred to below.

Softcover $34.95, ISBN 0-387-98976-5
Hardcover $69.95, ISBN 0-387-95014-1

Further information (reviews, errata, free copies for lecturers etc.) and electronic order forms can be found on

http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/
http://www.springer-ny.com/supplements/diestel/

Preface

Almost two decades have passed since the appearance of those graph theory texts that still set the agenda for most introductory courses taught today. The canon created by those books has helped to identify some main fields of study and research, and will doubtless continue to influence the development of the discipline for some time to come.

Yet much has happened in those 20 years, in graph theory no less than elsewhere: deep new theorems have been found, seemingly disparate methods and results have become interrelated, entire new branches have arisen. To name just a few such developments, one may think of how the new notion of list colouring has bridged the gulf between invariants such as average degree and chromatic number, how probabilistic methods and the regularity lemma have pervaded extremal graph theory and Ramsey theory, or how the entirely new field of graph minors and tree-decompositions has brought standard methods of surface topology to bear on long-standing algorithmic graph problems.

Clearly, then, the time has come for a reappraisal: *what are, today, the essential areas, methods and results that should form the centre of an introductory graph theory course aiming to equip its audience for the most likely developments ahead?*

I have tried in this book to offer material for such a course. In view of the increasing complexity and maturity of the subject, I have broken with the tradition of attempting to cover both theory and applications: this book offers an introduction to the theory of graphs as part of (pure) mathematics; it contains neither explicit algorithms nor 'real world' applications. My hope is that the potential for depth gained by this restriction in scope will serve students of computer science as much as their peers in mathematics: assuming that they prefer algorithms but will benefit from an encounter with pure mathematics of *some* kind, it seems an ideal opportunity to look for this close to where their heart lies!

In the selection and presentation of material, I have tried to accommodate two conflicting goals. On the one hand, I believe that an introductory text should be lean and concentrate on the essential, so as to offer guidance to those new to the field. As a graduate text, moreover, it should get to the heart of the matter quickly: after all, the idea is to convey at least an impression of the depth and methods of the subject. On the other hand, it has been my particular concern to write with sufficient detail to make the text enjoyable and easy to read: guiding questions and ideas will be discussed explicitly, and all proofs presented will be rigorous and complete.

A typical chapter, therefore, begins with a brief discussion of what are the guiding questions in the area it covers, continues with a succinct account of its classic results (often with simplified proofs), and then presents one or two deeper theorems that bring out the full flavour of that area. The proofs of these latter results are typically preceded by (or interspersed with) an informal account of their main ideas, but are then presented formally at the same level of detail as their simpler counterparts. I soon noticed that, as a consequence, some of those proofs came out rather longer in print than seemed fair to their often beautifully simple conception. I would hope, however, that even for the professional reader the relatively detailed account of those proofs will at least help to minimize reading time...

If desired, this text can be used for a lecture course with little or no further preparation. The simplest way to do this would be to follow the order of presentation, chapter by chapter: apart from two clearly marked exceptions, any results used in the proof of others precede them in the text.

Alternatively, a lecturer may wish to divide the material into an easy basic course for one semester, and a more challenging follow-up course for another. To help with the preparation of courses deviating from the order of presentation, I have listed in the margin next to each proof the reference numbers of those results that are used in that proof. These references are given in round brackets: for example, a reference (4.1.2) in the margin next to the proof of Theorem 4.3.2 indicates that Lemma 4.1.2 will be used in this proof. Correspondingly, in the margin next to Lemma 4.1.2 there is a reference [4.3.2] (in square brackets) informing the reader that this lemma will be used in the proof of Theorem 4.3.2. Note that this system applies between different sections only (of the same or of different chapters): the sections themselves are written as units and best read in their order of presentation.

The mathematical prerequisites for this book, as for most graph theory texts, are minimal: a first grounding in linear algebra is assumed for Chapter 1.9 and once in Chapter 5.5, some basic topological concepts about the Euclidean plane and 3-space are used in Chapter 4, and a previous first encounter with elementary probability will help with Chapter 11. (Even here, all that is assumed formally is the knowledge of basic definitions: the few probabilistic tools used are developed in the text.) There are two areas of graph theory which I find both fascinating and important, especially from the perspective of pure mathematics adopted here, but which are not covered in this book: these are algebraic graph theory and infinite graphs.

At the end of each chapter, there is a section with exercises and another with bibliographical and historical notes. Many of the exercises were chosen to complement the main narrative of the text: they illustrate new concepts, show how a new invariant relates to earlier ones, or indicate ways in which a result stated in the text is best possible. Particularly easy exercises are identified by the superscript $^-$, the more challenging ones carry a $^+$. The notes are intended to guide the reader on to further reading, in particular to any monographs or survey articles on the theme of that chapter. They also offer some historical and other remarks on the material presented in the text.

Ends of proofs are marked by the symbol □. Where this symbol is found directly below a formal assertion, it means that the proof should be clear after what has been said—a claim waiting to be verified! There are also some deeper theorems which are stated, without proof, as background information: these can be identified by the absence of both proof and □.

Almost every book contains errors, and this one will hardly be an exception. I shall try to post on the Web any corrections that become necessary. The relevant site may change in time, but will always be accessible via the following two addresses:

http://www.springer-ny.com/supplements/diestel/
http://www.springer.de/catalog/html-files/deutsch/math/3540609180.html

Please let me know about any errors you find.

Little in a textbook is truly original: even the style of writing and of presentation will invariably be influenced by examples. The book that no doubt influenced me most is the classic GTM graph theory text by Bollobás: it was in the course recorded by this text that I learnt my first graph theory as a student. Anyone who knows this book well will feel its influence here, despite all differences in contents and presentation.

I should like to thank all who gave so generously of their time, knowledge and advice in connection with this book. I have benefited particularly from the help of N. Alon, G. Brightwell, R. Gillett, R. Halin, M. Hintz, A. Huck, I. Leader, T. Łuczak, W. Mader, V. Rödl, A.D. Scott, P.D. Seymour, G. Simonyi, M. Škoviera, R. Thomas, C. Thomassen and P. Valtr. I am particularly grateful also to Tommy R. Jensen, who taught me much about colouring and all I know about k-flows, and who invested immense amounts of diligence and energy in his proofreading of the preliminary German version of this book.

March 1997 *RD*

About the second edition

Naturally, I am delighted at having to write this addendum so soon after this book came out in the summer of 1997. It is particularly gratifying to hear that people are gradually adopting it not only for their personal use but more and more also as a course text; this, after all, was my aim when I wrote it, and my excuse for agonizing more over presentation than I might otherwise have done.

There are two major changes. The last chapter on graph minors now gives a complete proof of one of the major results of the Robertson-Seymour theory, their theorem that excluding a graph as a minor bounds the tree-width if and only if that graph is planar. This short proof did not exist when I wrote the first edition, which is why I then included a short proof of the next best thing, the analogous result for path-width. That theorem has now been dropped from Chapter 12. Another addition in this chapter is that the tree-width duality theorem, Theorem 12.3.9, now comes with a (short) proof too.

The second major change is the addition of a complete set of hints for the exercises. These are largely Tommy Jensen's work, and I am grateful for the time he donated to this project. The aim of these hints is to help those who use the book to study graph theory on their own, but *not* to spoil the fun. The exercises, including hints, continue to be intended for classroom use.

Apart from these two changes, there are a few additions. The most noticable of these are the formal introduction of depth-first search trees in Section 1.5 (which has led to some simplifications in later proofs) and an ingenious new proof of Menger's theorem due to Böhme, Göring and Harant (which has not otherwise been published).

Finally, there is a host of small simplifications and clarifications of arguments that I noticed as I taught from the book, or which were pointed out to me by others. To all these I offer my special thanks.

The Web site for the book has followed me to

http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/

I expect this address to be stable for some time.

Once more, my thanks go to all who contributed to this second edition by commenting on the first—and I look forward to further comments!

December 1999 *RD*

Contents

Preface ... vii

1. The Basics ... 1

1.1. Graphs ... 2
1.2. The degree of a vertex ... 4
1.3. Paths and cycles ... 6
1.4. Connectivity ... 9
1.5. Trees and forests ... 12
1.6. Bipartite graphs ... 14
1.7. Contraction and minors ... 16
1.8. Euler tours ... 18
1.9. Some linear algebra ... 20
1.10. Other notions of graphs ... 25
Exercises ... 26
Notes ... 28

2. Matching ... 29

2.1. Matching in bipartite graphs ... 29
2.2. Matching in general graphs ... 34
2.3. Path covers ... 39
Exercises ... 40
Notes ... 42

3. Connectivity 43
3.1. 2-Connected graphs and subgraphs 43
3.2. The structure of 3-connected graphs 45
3.3. Menger's theorem 50
3.4. Mader's theorem 56
3.5. Edge-disjoint spanning trees 58
3.6. Paths between given pairs of vertices 61
Exercises 63
Notes 65

4. Planar Graphs 67
4.1. Topological prerequisites 68
4.2. Plane graphs 70
4.3. Drawings 76
4.4. Planar graphs: Kuratowski's theorem 80
4.5. Algebraic planarity criteria 85
4.6. Plane duality 87
Exercises 89
Notes 92

5. Colouring 95
5.1. Colouring maps and planar graphs 96
5.2. Colouring vertices 98
5.3. Colouring edges 103
5.4. List colouring 105
5.5. Perfect graphs 110
Exercises 117
Notes 120

6. Flows 123
6.1. Circulations 124
6.2. Flows in networks 125
6.3. Group-valued flows 128
6.4. k-Flows for small k 133
6.5. Flow-colouring duality 136
6.6. Tutte's flow conjectures 140
Exercises 144
Notes 145

7. Substructures in Dense Graphs ... 147
7.1. Subgraphs ... 148
7.2. Szemerédi's regularity lemma ... 153
7.3. Applying the regularity lemma ... 160
Exercises ... 165
Notes ... 166

8. Substructures in Sparse Graphs ... 169
8.1. Topological minors ... 170
8.2. Minors ... 179
8.3. Hadwiger's conjecture ... 181
Exercises ... 184
Notes ... 186

9. Ramsey Theory for Graphs ... 189
9.1. Ramsey's original theorems ... 190
9.2. Ramsey numbers ... 193
9.3. Induced Ramsey theorems ... 197
9.4. Ramsey properties and connectivity ... 207
Exercises ... 208
Notes ... 210

10. Hamilton Cycles ... 213
10.1. Simple sufficient conditions ... 213
10.2. Hamilton cycles and degree sequences ... 216
10.3. Hamilton cycles in the square of a graph ... 218
Exercises ... 226
Notes ... 227

11. Random Graphs ... 229
11.1. The notion of a random graph ... 230
11.2. The probabilistic method ... 235
11.3. Properties of almost all graphs ... 238
11.4. Threshold functions and second moments ... 242
Exercises ... 247
Notes ... 249

12. Minors, Trees, and WQO 251
12.1. Well-quasi-ordering 251
12.2. The graph minor theorem for trees 253
12.3. Tree-decompositions 255
12.4. Tree-width and forbidden minors 263
12.5. The graph minor theorem 274
Exercises 277
Notes 280

Hints for all the exercises 283
Index 299
Symbol index 311

1 The Basics

This chapter gives a gentle yet concise introduction to most of the terminology used later in the book. Fortunately, much of standard graph theoretic terminology is so intuitive that it is easy to remember; the few terms better understood in their proper setting will be introduced later, when their time has come.

Section 1.1 offers a brief but self-contained summary of the most basic definitions in graph theory, those centred round the notion of a graph. Most readers will have met these definitions before, or will have them explained to them as they begin to read this book. For this reason, Section 1.1 does not dwell on these definitions more than clarity requires: its main purpose is to collect the most basic terms in one place, for easy reference later.

From Section 1.2 onwards, all new definitions will be brought to life almost immediately by a number of simple yet fundamental propositions. Often, these will relate the newly defined terms to one another: the question of how the value of one invariant influences that of another underlies much of graph theory, and it will be good to become familiar with this line of thinking early.

By $\mathbb{N}$ we denote the set of natural numbers, including zero. The set $\mathbb{Z}/n\mathbb{Z}$ of integers modulo n is denoted by $\mathbb{Z}_n$; its elements are written as $\mathbb{Z}_n$
$\overline{i} := i + n\mathbb{Z}$. For a real number x we denote by $\lfloor x \rfloor$ the greatest integer $\leqslant x$, and by $\lceil x \rceil$ the least integer $\geqslant x$. Logarithms written as 'log' are $\lfloor x \rfloor, \lceil x \rceil$
taken at base 2; the natural logarithm will be denoted by 'ln'. A set log, ln
$\mathcal{A} = \{ A_1, \dots, A_k \}$ of disjoint subsets of a set A is a *partition* of A if *partition*
$A = \bigcup_{i=1}^k A_i$ and $A_i \neq \emptyset$ for every i. Another partition $\{ A'_1, \dots, A'_\ell \}$ of A *refines* the partition $\mathcal{A}$ if each A'_i is contained in some A_j. By $[A]^k$ we $[A]^k$
denote the set of all k-element subsets of A. Sets with k elements will be called *k-sets*; subsets with k elements are *k-subsets*. *k-set*

1.1 Graphs

graph · *vertex* · *edge*

A *graph* is a pair $G = (V, E)$ of sets satisfying $E \subseteq [V]^2$; thus, the elements of E are 2-element subsets of V. To avoid notational ambiguities, we shall always assume tacitly that $V \cap E = \emptyset$. The elements of V are the *vertices* (or *nodes*, or *points*) of the graph G, the elements of E are its *edges* (or *lines*). The usual way to picture a graph is by drawing a dot for each vertex and joining two of these dots by a line if the corresponding two vertices form an edge. Just how these dots and lines are drawn is considered irrelevant: all that matters is the information which pairs of vertices form an edge and which do not.

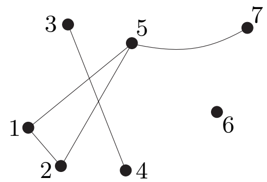

Fig. 1.1.1. The graph on $V = \{1, \ldots, 7\}$ with edge set $E = \{\{1,2\}, \{1,5\}, \{2,5\}, \{3,4\}, \{5,7\}\}$

on · *$V(G), E(G)$*

A graph with vertex set V is said to be a graph *on* V. The vertex set of a graph G is referred to as $V(G)$, its edge set as $E(G)$. These conventions are independent of any actual names of these two sets: the vertex set W of a graph $H = (W, F)$ is still referred to as $V(H)$, not as $W(H)$. We shall not always distinguish strictly between a graph and its vertex or edge set. For example, we may speak of a vertex $v \in G$ (rather than $v \in V(G)$), an edge $e \in G$, and so on.

order · $|G|$, $\|G\|$

The number of vertices of a graph G is its *order*, written as $|G|$; its number of edges is denoted by $\|G\|$. Graphs are *finite* or *infinite* according to their order; unless otherwise stated, the graphs we consider are all finite.

$\emptyset$ · *trivial graph*

For the *empty graph* $(\emptyset, \emptyset)$ we simply write $\emptyset$. A graph of order 0 or 1 is called *trivial*. Sometimes, e.g. to start an induction, trivial graphs can be useful; at other times they form silly counterexamples and become a nuisance. To avoid cluttering the text with non-triviality conditions, we shall mostly treat the trivial graphs, and particularly the empty graph $\emptyset$, with generous disregard.

incident · *ends* · $E(X,Y)$ · $E(v)$

A vertex v is *incident* with an edge e if $v \in e$; then e is an edge *at* v. The two vertices incident with an edge are its *endvertices* or *ends*, and an edge *joins* its ends. An edge $\{x, y\}$ is usually written as xy (or yx). If $x \in X$ and $y \in Y$, then xy is an X–Y *edge*. The set of all X–Y edges in a set E is denoted by $E(X,Y)$; instead of $E(\{x\}, Y)$ and $E(X, \{y\})$ we simply write $E(x, Y)$ and $E(X, y)$. The set of all the edges in E at a vertex v is denoted by $E(v)$.

Two vertices x, y of G are *adjacent*, or *neighbours*, if xy is an edge of G. Two edges $e \neq f$ are *adjacent* if they have an end in common. If all the vertices of G are pairwise adjacent, then G is *complete*. A complete graph on n vertices is a K^n; a K^3 is called a *triangle*.

adjacent *neighbour* *complete* K^n

Pairwise non-adjacent vertices or edges are called *independent*. More formally, a set of vertices or of edges is *independent* (or *stable*) if no two of its elements are adjacent.

inde-pendent

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. We call G and G' *isomorphic*, and write $G \simeq G'$, if there exists a bijection $\varphi: V \to V'$ with $xy \in E \Leftrightarrow \varphi(x)\varphi(y) \in E'$ for all $x, y \in V$. Such a map φ is called an *isomorphism*; if $G = G'$, it is called an *automorphism*. We do not normally distinguish between isomorphic graphs. Thus, we usually write $G = G'$ rather than $G \simeq G'$, speak of *the* complete graph on 17 vertices, and so on. A map taking graphs as arguments is called a *graph invariant* if it assigns equal values to isomorphic graphs. The number of vertices and the number of edges of a graph are two simple graph invariants; the greatest number of pairwise adjacent vertices is another.

$\simeq$ *isomor-phism* *invariant*

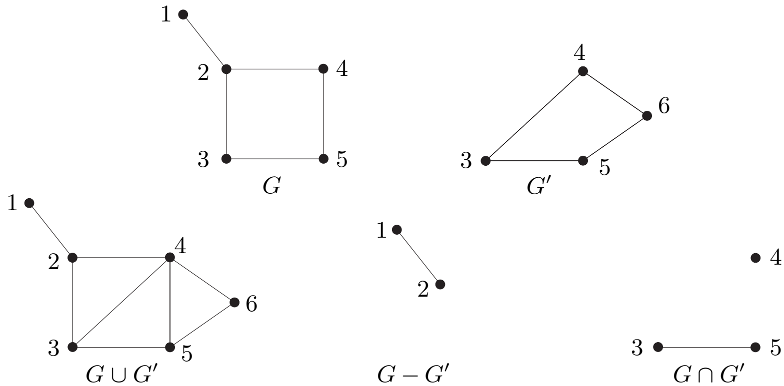

Fig. 1.1.2. Union, difference and intersection; the vertices 2,3,4 induce (or span) a triangle in $G \cup G'$ but not in G

We set $G \cup G' := (V \cup V', E \cup E')$ and $G \cap G' := (V \cap V', E \cap E')$. If $G \cap G' = \emptyset$, then G and G' are *disjoint*. If $V' \subseteq V$ and $E' \subseteq E$, then G' is a *subgraph* of G (and G a *supergraph* of G'), written as $G' \subseteq G$. Less formally, we say that G *contains* G'.

$G \cap G'$ *subgraph* $G' \subseteq G$

If $G' \subseteq G$ and G' contains *all* the edges $xy \in E$ with $x, y \in V'$, then G' is an *induced subgraph* of G; we say that V' *induces* or *spans* G' in G, and write $G' =: G[V']$. Thus if $U \subseteq V$ is any set of vertices, then $G[U]$ denotes the graph on U whose edges are precisely the edges of G with both ends in U. If H is a subgraph of G, not necessarily induced, we abbreviate $G[V(H)]$ to $G[H]$. Finally, $G' \subseteq G$ is a *spanning* subgraph of G if V' spans all of G, i.e. if $V' = V$.

induced subgraph $G[U]$ *spanning*

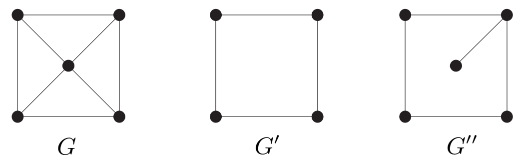

Fig. 1.1.3. A graph G with subgraphs G' and G'': G' is an induced subgraph of G, but G'' is not

$-$

If U is any set of vertices (usually of G), we write $G - U$ for $G[V \setminus U]$. In other words, $G - U$ is obtained from G by *deleting* all the vertices in $U \cap V$ and their incident edges. If $U = \{v\}$ is a singleton, we write $G - v$ rather than $G - \{v\}$. Instead of $G - V(G')$ we simply write $G - G'$. For a subset F of $[V]^2$ we write $G - F := (V, E \setminus F)$ and $G + F := (V, E \cup F)$; as above, $G - \{e\}$ and $G + \{e\}$ are abbreviated to $G - e$ and $G + e$. We call G *edge-maximal* with a given graph property if G itself has the property but no graph $G + xy$ does, for non-adjacent vertices $x, y \in G$.

$+$ *edge-maximal*

minimal *maximal*

More generally, when we call a graph *minimal* or *maximal* with some property but have not specified any particular ordering, we are referring to the subgraph relation. When we speak of minimal or maximal sets of vertices or edges, the reference is simply to set inclusion.

$G * G'$ *complement $\overline{G}$* *line graph $L(G)$*

If G and G' are disjoint, we denote by $G * G'$ the graph obtained from $G \cup G'$ by joining all the vertices of G to all the vertices of G'. For example, $K^2 * K^3 = K^5$. The *complement* $\overline{G}$ of G is the graph on V with edge set $[V]^2 \setminus E$. The *line graph* $L(G)$ of G is the graph on E in which $x, y \in E$ are adjacent as vertices if and only if they are adjacent as edges in G.

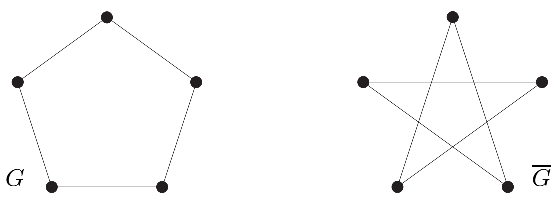

Fig. 1.1.4. A graph isomorphic to its complement

1.2 The degree of a vertex

$N(v)$

Let $G = (V, E)$ be a (non-empty) graph. The set of neighbours of a vertex v in G is denoted by $N_G(v)$, or briefly by $N(v)$.[1] More generally

[1] Here, as elsewhere, we drop the index referring to the underlying graph if the reference is clear.

for $U \subseteq V$, the neighbours in $V \setminus U$ of vertices in U are called *neighbours of* U; their set is denoted by $N(U)$.

The *degree* (or *valency*) $d_G(v) = d(v)$ of a vertex v is the number $|E(v)|$ of edges at v; by our definition of a graph,[2] this is equal to the number of neighbours of v. A vertex of degree 0 is *isolated*. The number $\delta(G) := \min\{\, d(v) \mid v \in V \,\}$ is the *minimum degree* of G, the number $\Delta(G) := \max\{\, d(v) \mid v \in V \,\}$ its *maximum degree*. If all the vertices of G have the same degree k, then G is k-*regular*, or simply *regular*. A 3-regular graph is called *cubic*. *degree* $d(v)$ *isolated* $\delta(G)$ $\Delta(G)$ *regular* *cubic*

The number

$$d(G) := \frac{1}{|V|} \sum_{v \in V} d(v)$$

$d(G)$

is the *average degree* of G. Clearly, *average degree*

$$\delta(G) \leqslant d(G) \leqslant \Delta(G)\,.$$

The average degree quantifies globally what is measured locally by the vertex degrees: the number of edges of G per vertex. Sometimes it will be convenient to express this ratio directly, as $\varepsilon(G) := |E|/|V|$. $\varepsilon(G)$

The quantities d and ε are, of course, intimately related. Indeed, if we sum up all the vertex degrees in G, we count every edge exactly twice: once from each of its ends. Thus

$$|E| = \tfrac{1}{2} \sum_{v \in V} d(v) = \tfrac{1}{2} d(G) \cdot |V|\,,$$

and therefore

$$\varepsilon(G) = \tfrac{1}{2} d(G)\,.$$

Proposition 1.2.1. *The number of vertices of odd degree in a graph is always even.* [10.3.3]

Proof. A graph on V has $\frac{1}{2}\sum_{v \in V} d(v)$ edges, so $\sum d(v)$ is an even number. □

If a graph has large minimum degree, i.e. everywhere, locally, many edges per vertex, it also has many edges per vertex globally: $\varepsilon(G) = \frac{1}{2}d(G) \geqslant \frac{1}{2}\delta(G)$. Conversely, of course, its average degree may be large even when its minimum degree is small. However, the vertices of large degree cannot be scattered completely among vertices of small degree: as the next proposition shows, every graph G has a subgraph whose average degree is no less than the average degree of G, and whose minimum degree is more than half its average degree:

[2] but not for multigraphs; see Section 1.10

[3.6.1] **Proposition 1.2.2.** *Every graph G with at least one edge has a subgraph H with $\delta(H) > \varepsilon(H) \geqslant \varepsilon(G)$.*

Proof. To construct H from G, let us try to delete vertices of small degree one by one, until only vertices of large degree remain. Up to which degree $d(v)$ can we afford to delete a vertex v, without lowering ε? Clearly, up to $d(v) = \varepsilon$: then the number of vertices decreases by 1 and the number of edges by at most ε, so the overall ratio ε of edges to vertices will not decrease.

Formally, we construct a sequence $G = G_0 \supseteq G_1 \supseteq \ldots$ of induced subgraphs of G as follows. If G_i has a vertex v_i of degree $d(v_i) \leqslant \varepsilon(G_i)$, we let $G_{i+1} := G_i - v_i$; if not, we terminate our sequence and set $H := G_i$. By the choices of v_i we have $\varepsilon(G_{i+1}) \geqslant \varepsilon(G_i)$ for all i, and hence $\varepsilon(H) \geqslant \varepsilon(G)$.

What else can we say about the graph H? Since $\varepsilon(K^1) = 0 < \varepsilon(G)$, none of the graphs in our sequence is trivial, so in particular $H \neq \emptyset$. The fact that H has no vertex suitable for deletion thus implies $\delta(H) > \varepsilon(H)$, as claimed. □

1.3 Paths and cycles

path

A *path* is a non-empty graph $P = (V, E)$ of the form

$$V = \{ x_0, x_1, \ldots, x_k \} \qquad E = \{ x_0x_1, x_1x_2, \ldots, x_{k-1}x_k \},$$

where the x_i are all distinct. The vertices x_0 and x_k are *linked* by P and are called its *ends*; the vertices $x_1, \ldots, x_{k-1}$ are the *inner* vertices of P.

length
P^k

The number of edges of a path is its *length*, and the path of length k is denoted by P^k. Note that k is allowed to be zero; thus, $P^0 = K^1$.

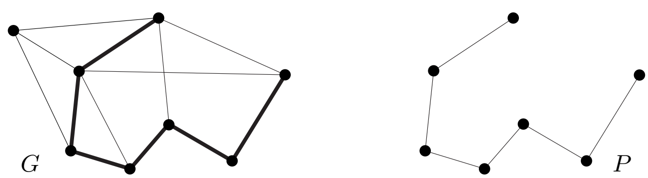

Fig. 1.3.1. A path $P = P^6$ in G

We often refer to a path by the natural sequence of its vertices,[3] writing, say, $P = x_0x_1 \ldots x_k$ and calling P a path *from* x_0 *to* x_k (as well as *between* x_0 and x_k).

[3] More precisely, by one of the two natural sequences: $x_0 \ldots x_k$ and $x_k \ldots x_0$ denote the same path. Still, it often helps to fix one of these two orderings of $V(P)$ notationally: we may then speak of things like the 'first' vertex on P with a certain property, etc.

xPy, $\mathring{P}$

For $0 \leqslant i \leqslant j \leqslant k$ we write

$$\begin{aligned} Px_i &:= x_0 \dots x_i \\ x_iP &:= x_i \dots x_k \\ x_iPx_j &:= x_i \dots x_j \end{aligned}$$

and

$$\begin{aligned} \mathring{P} &:= x_1 \dots x_{k-1} \\ P\mathring{x}_i &:= x_0 \dots x_{i-1} \\ \mathring{x}_iP &:= x_{i+1} \dots x_k \\ \mathring{x}_iP\mathring{x}_j &:= x_{i+1} \dots x_{j-1} \end{aligned}$$

for the appropriate subpaths of P. We use similar intuitive notation for the concatenation of paths; for example, if the union $Px \cup xQy \cup yR$ of three paths is again a path, we may simply denote it by $PxQyR$.

$PxQyR$

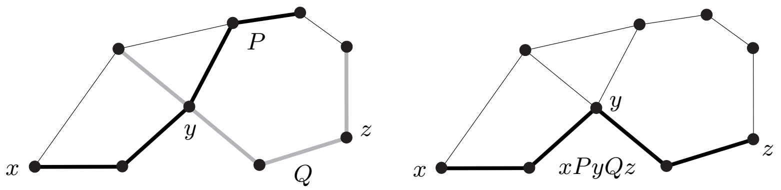

Fig. 1.3.2. Paths P, Q and $xPyQz$

A–B path

independent

Given sets A, B of vertices, we call $P = x_0 \dots x_k$ an *A–B path* if $V(P) \cap A = \{ x_0 \}$ and $V(P) \cap B = \{ x_k \}$. As before, we write a–B path rather than $\{ a \}$–B path, etc. Two or more paths are *independent* if none of them contains an inner vertex of another. Two a–b paths, for instance, are independent if and only if a and b are their only common vertices.

H-path

Given a graph H, we call P an *H-path* if P is non-trivial and meets H exactly in its ends. In particular, the edge of any H-path of length 1 is never an edge of H.

cycle

length

C^k

If $P = x_0 \dots x_{k-1}$ is a path and $k \geqslant 3$, then the graph $C := P + x_{k-1}x_0$ is called a *cycle*. As with paths, we often denote a cycle by its (cyclic) sequence of vertices; the above cycle C might be written as $x_0 \dots x_{k-1}x_0$. The *length* of a cycle is its number of edges (or vertices); the cycle of length k is called a *k-cycle* and denoted by C^k.

girth $g(G)$

circumference

chord

induced cycle

The minimum length of a cycle (contained) in a graph G is the *girth* $g(G)$ of G; the maximum length of a cycle in G is its *circumference*. (If G does not contain a cycle, we set the former to ∞, the latter to zero.) An edge which joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. Thus, an *induced cycle* in G, a cycle in G forming an induced subgraph, is one that has no chords (Fig. 1.3.3).

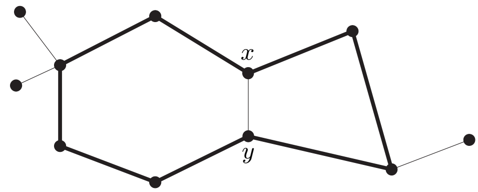

Fig. 1.3.3. A cycle C^8 with chord xy, and induced cycles C^6, C^4

If a graph has large minimum degree, it contains long paths and cycles:

[3.6.1] **Proposition 1.3.1.** *Every graph G contains a path of length $\delta(G)$ and a cycle of length at least $\delta(G)+1$ (provided that $\delta(G) \geqslant 2$).*

Proof. Let $x_0 \dots x_k$ be a longest path in G. Then all the neighbours of x_k lie on this path (Fig. 1.3.4). Hence $k \geqslant d(x_k) \geqslant \delta(G)$. If $i < k$ is minimal with $x_i x_k \in E(G)$, then $x_i \dots x_k x_i$ is a cycle of length at least $\delta(G)+1$. □

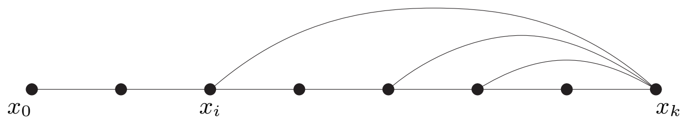

Fig. 1.3.4. A longest path $x_0 \dots x_k$, and the neighbours of x_k

Minimum degree and girth, on the other hand, are not related (unless we fix the number of vertices): as we shall see in Chapter 11, there are graphs combining arbitrarily large minimum degree with arbitrarily large girth.

distance $d_G(x,y)$

The *distance* $d_G(x,y)$ in G of two vertices x,y is the length of a shortest x–y path in G; if no such path exists, we set $d(x,y) := \infty$. The greatest distance between any two vertices in G is the *diameter* of G, denoted by $\mathrm{diam}(G)$. Diameter and girth are, of course, related:

diameter $\mathrm{diam}(G)$

Proposition 1.3.2. *Every graph G containing a cycle satisfies $g(G) \leqslant 2\,\mathrm{diam}(G)+1$.*

Proof. Let C be a shortest cycle in G. If $g(G) \geqslant 2\,\mathrm{diam}(G)+2$, then C has two vertices whose distance in C is at least $\mathrm{diam}(G)+1$. In G, these vertices have a lesser distance; any shortest path P between them is therefore not a subgraph of C. Thus, P contains a C-path xPy. Together with the shorter of the two x–y paths in C, this path xPy forms a shorter cycle than C, a contradiction. □

A vertex is *central* in G if its greatest distance from any other vertex is as small as possible. This distance is the *radius* of G, denoted by $\mathrm{rad}(G)$. Thus, formally, $\mathrm{rad}(G) = \min_{x \in V(G)} \max_{y \in V(G)} d_G(x,y)$. As one easily checks (exercise), we have *central* *radius* $\mathrm{rad}(G)$

$$\mathrm{rad}(G) \leqslant \mathrm{diam}(G) \leqslant 2\,\mathrm{rad}(G)\,.$$

Diameter and radius are not directly related to the minimum or average degree: a graph can combine large minimum degree with large diameter, or small average degree with small diameter (examples?).

The maximum degree behaves differently here: a graph of large order can only have small radius and diameter if its maximum degree is large. This connection is quantified very roughly in the following proposition:

Proposition 1.3.3. *A graph G of radius at most k and maximum degree at most d has no more than $1 + kd^k$ vertices.* [9.4.1] [9.4.2]

Proof. Let z be a central vertex in G, and let D_i denote the set of vertices of G at distance i from z. Then $V(G) = \bigcup_{i=0}^{k} D_i$, and $|D_0| = 1$. Since $\Delta(G) \leqslant d$, we have $|D_i| \leqslant d\,|D_{i-1}|$ for $i = 1,\dots,k$, and thus $|D_i| \leqslant d^i$ by induction. Adding up these inequalities we obtain

$$|G| \leqslant 1 + \sum_{i=1}^{k} d^i \leqslant 1 + kd^k.$$

□

A *walk* (of *length* k) in a graph G is a non-empty alternating sequence $v_0 e_0 v_1 e_1 \dots e_{k-1} v_k$ of vertices and edges in G such that $e_i = \{\, v_i, v_{i+1} \,\}$ for all $i < k$. If $v_0 = v_k$, the walk is *closed*. If the vertices in a walk are all distinct, it defines an obvious path in G. In general, every walk between two vertices contains[4] a path between these vertices (proof?). *walk*

1.4 Connectivity

A non-empty graph G is called *connected* if any two of its vertices are linked by a path in G. If $U \subseteq V(G)$ and $G[\,U\,]$ is connected, we also call U itself connected (in G). *connected*

Proposition 1.4.1. *The vertices of a connected graph G can always be enumerated, say as $v_1,\dots,v_n$, so that $G_i := G[\,v_1,\dots,v_i\,]$ is connected for every i.* [1.5.2]

[4] We shall often use terms defined for graphs also for walks, as long as their meaning is obvious.

Proof. Pick any vertex as v_1, and assume inductively that $v_1, \dots, v_i$ have been chosen for some $i < |G|$. Now pick a vertex $v \in G - G_i$. As G is connected, it contains a v–v_1 path P. Choose as v_{i+1} the last vertex of P in $G - G_i$; then v_{i+1} has a neighbour in G_i. The connectedness of every G_i follows by induction on i. □

component

Let $G = (V, E)$ be a graph. A maximal connected subgraph of G is called a *component* of G. Note that a component, being connected, is always non-empty; the empty graph, therefore, has no components.

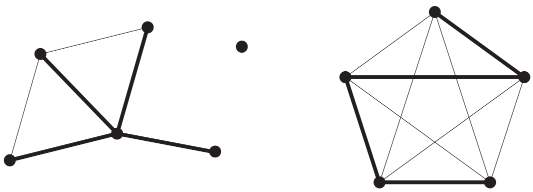

Fig. 1.4.1. A graph with three components, and a minimal spanning connected subgraph in each component

separate

cutvertex

bridge

If $A, B \subseteq V$ and $X \subseteq V \cup E$ are such that every A–B path in G contains a vertex or an edge from X, we say that X *separates* the sets A and B in G. This implies in particular that $A \cap B \subseteq X$. More generally we say that X *separates* G, and call X a *separating set* in G, if X separates two vertices of $G - X$ in G. A vertex which separates two other vertices of the same component is a *cutvertex*, and an edge separating its ends is a *bridge*. Thus, the bridges in a graph are precisely those edges that do not lie on any cycle.

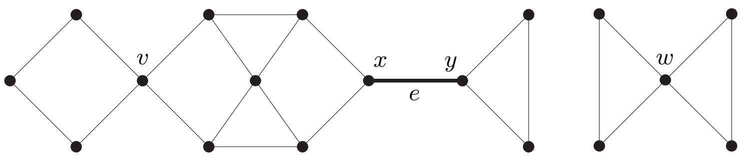

Fig. 1.4.2. A graph with cutvertices v, x, y, w and bridge $e = xy$

k-connected

G is called *k-connected* (for $k \in \mathbb{N}$) if $|G| > k$ and $G - X$ is connected for every set $X \subseteq V$ with $|X| < k$. In other words, no two vertices of G are separated by fewer than k other vertices. Every (non-empty) graph is 0-connected, and the 1-connected graphs are precisely the non-trivial connected graphs. The greatest integer k such that G is k-connected is the *connectivity* $\kappa(G)$ of G. Thus, $\kappa(G) = 0$ if and only if G is disconnected or a K^1, and $\kappa(K^n) = n - 1$ for all $n \geqslant 1$.

connectivity $\kappa(G)$

ℓ-edge-connected

If $|G| > 1$ and $G - F$ is connected for every set $F \subseteq E$ of fewer than ℓ edges, then G is called *ℓ-edge-connected.* The greatest integer ℓ

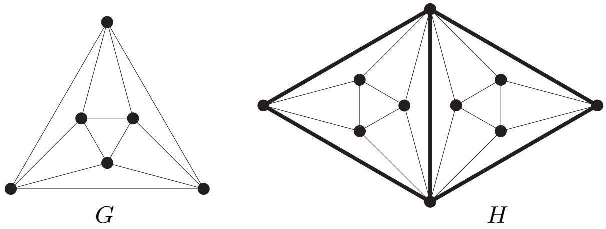

Fig. 1.4.3. The *octahedron* G (left) with $\kappa(G) = \lambda(G) = 4$, and a graph H with $\kappa(H) = 2$ but $\lambda(H) = 4$

such that G is ℓ-edge-connected is the *edge-connectivity* $\lambda(G)$ of G. In particular, we have $\lambda(G) = 0$ if G is disconnected. *edge-connectivity* $\lambda(G)$

For every non-trivial graph G we have

$$\kappa(G) \leqslant \lambda(G) \leqslant \delta(G)$$

(exercise), so in particular high connectivity requires a large minimum degree. Conversely, large minimum degree does not ensure high connectivity, not even high edge-connectivity (examples?). It does, however, imply the existence of a highly connected subgraph:

Theorem 1.4.2. (Mader 1972)
Every graph of average degree at least $4k$ has a k-connected subgraph. [8.1.1] [11.2.3]

Proof. For $k \in \{0, 1\}$ the assertion is trivial; we consider $k \geqslant 2$ and a graph $G = (V, E)$ with $|V| =: n$ and $|E| =: m$. For inductive reasons it will be easier to prove the stronger assertion that G has a k-connected subgraph whenever

(i) $n \geqslant 2k - 1$ and

(ii) $m \geqslant (2k-3)(n-k+1) + 1$.

(This assertion is indeed stronger, i.e. (i) and (ii) follow from our assumption of $d(G) \geqslant 4k$: (i) holds since $n > \Delta(G) \geqslant d(G) \geqslant 4k$, while (ii) follows from $m = \frac{1}{2}d(G)n \geqslant 2kn$.)

We apply induction on n. If $n = 2k-1$, then $k = \frac{1}{2}(n+1)$, and hence $m \geqslant \frac{1}{2}n(n-1)$ by (ii). Thus $G = K^n \supseteq K^{k+1}$, proving our claim. We now assume that $n \geqslant 2k$. If v is a vertex with $d(v) \leqslant 2k-3$, we can apply the induction hypothesis to $G - v$ and are done. So we assume that $\delta(G) \geqslant 2k-2$. If G is k-connected, there is nothing to show. We may therefore assume that G has the form $G = G_1 \cup G_2$ with $|G_1 \cap G_2| < k$ and $|G_1|, |G_2| < n$. As every edge of G lies in G_1 or in G_2, G has no edge between $G_1 - G_2$ and $G_2 - G_1$. Since each vertex in these subgraphs has at least $\delta(G) \geqslant 2k-2$ neighbours, we have $|G_1|, |G_2| \geqslant 2k-1$. But then at least one of the graphs G_1, G_2 must satisfy the induction hypothesis (completing the proof): if neither does, we have

$$\|G_i\| \leqslant (2k-3)(|G_i| - k + 1)$$

for $i = 1, 2$, and hence

$$\begin{aligned} m &\leqslant \|G_1\| + \|G_2\| \\ &\leqslant (2k-3)\big(|G_1| + |G_2| - 2k + 2\big) \\ &\leqslant (2k-3)(n-k+1) \qquad (\text{by } |G_1 \cap G_2| \leqslant k-1) \end{aligned}$$

contradicting (ii). □

1.5 Trees and forests

forest *tree* *leaf*

An *acyclic* graph, one not containing any cycles, is called a *forest*. A connected forest is called a *tree*. (Thus, a forest is a graph whose components are trees.) The vertices of degree 1 in a tree are its *leaves*. Every non-trivial tree has at least two leaves—take, for example, the ends of a longest path. This little fact often comes in handy, especially in induction proofs about trees: if we remove a leaf from a tree, what remains is still a tree.

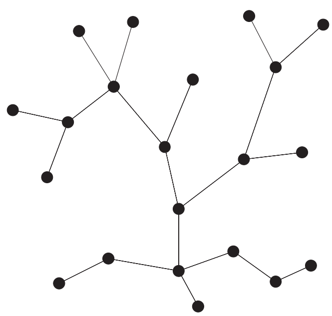

Fig. 1.5.1. A tree

[1.6.1] [1.9.6] [4.2.7]

Theorem 1.5.1. *The following assertions are equivalent for a graph T:*

(i) *T is a tree;*

(ii) *any two vertices of T are linked by a unique path in T;*

(iii) *T is minimally connected, i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$;*

(iv) *T is maximally acyclic, i.e. T contains no cycle but $T + xy$ does, for any two non-adjacent vertices $x, y \in T$.* □

The proof of Theorem 1.5.1 is straightforward, and a good exercise for anyone not yet familiar with all the notions it relates. Extending our notation for paths from Section 1.3, we write xTy for the unique path in a tree T between two vertices x, y (see (ii) above). *xTy*

A frequently used application of Theorem 1.5.1 is that every connected graph contains a spanning tree: by the equivalence of (i) and (iii), any minimal connected spanning subgraph will be a tree. Figure 1.4.1 shows a spanning tree in each of the three components of the graph depicted.

Corollary 1.5.2. *The vertices of a tree can always be enumerated, say as $v_1, \dots, v_n$, so that every v_i with $i \geqslant 2$ has a unique neighbour in $\{ v_1, \dots, v_{i-1} \}$.*

Proof. Use the enumeration from Proposition 1.4.1. □ (1.4.1)

[1.9.6] [3.5.1] [3.5.4] [4.2.7] [8.2.2]

Corollary 1.5.3. *A connected graph with n vertices is a tree if and only if it has $n-1$ edges.*

Proof. Induction on i shows that the subgraph spanned by the first i vertices in Corollary 1.5.2 has $i-1$ edges; for $i = n$ this proves the forward implication. Conversely, let G be any connected graph with n vertices and $n-1$ edges. Let G' be a spanning tree in G. Since G' has $n-1$ edges by the first implication, it follows that $G = G'$. □

[9.2.1] [9.2.3]

Corollary 1.5.4. *If T is a tree and G is any graph with $\delta(G) \geqslant |T|-1$, then $T \subseteq G$, i.e. G has a subgraph isomorphic to T.*

Proof. Find a copy of T in G inductively along its vertex enumeration from Corollary 1.5.2. □

Sometimes it is convenient to consider one vertex of a tree as special; such a vertex is then called the *root* of this tree. A tree with a fixed root is a *rooted tree*. Choosing a root r in a tree T imposes a partial ordering on $V(T)$ by letting $x \leqslant y$ if $x \in rTy$. This is the *tree-order* on $V(T)$ associated with T and r. Note that r is the least element in this partial order, every leaf $x \neq r$ of T is a maximal element, the ends of any edge of T are comparable, and every set of the form $\{ x \mid x \leqslant y \}$ (where y is any fixed vertex) is a *chain*, a set of pairwise comparable elements. (Proofs?) *root* *tree-order* *chain*

A rooted tree T contained in a graph G is called *normal* in G if the ends of every T-path in G are comparable in the tree-order of T. If T spans G, this amounts to requiring that two vertices of T must be comparable whenever they are adjacent in G; see Figure 1.5.2. Normal spanning trees are also called *depth-first search trees*, because of the way they arise in computer searches on graphs (Exercise 17). *normal tree*

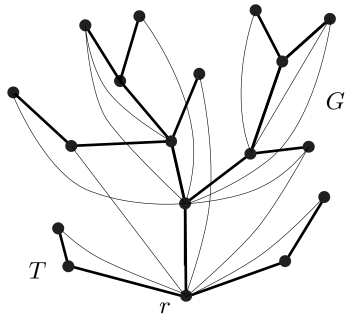

Fig. 1.5.2. A depth-first search tree with root r

Normal spanning trees provide a simple but powerful structural tool in graph theory. And they always exist:

[6.5.3] **Proposition 1.5.5.** *Every connected graph contains a normal spanning tree, with any specified vertex as its root.*

Proof. Let G be a connected graph and $r \in G$ any specified vertex. Let T be a maximal normal tree with root r in G; we show that $V(T) = V(G)$.

Suppose not, and let C be a component of $G - T$. As T is normal, $N(C)$ is a chain in T. Let x be its greatest element, and let $y \in C$ be adjacent to x. Let T' be the tree obtained from T by joining y to x; the tree-order of T' then extends that of T. We shall derive a contradiction by showing that T' is also normal in G.

Let P be a T'-path in G. If the ends of P both lie in T, then they are comparable in the tree-order of T (and hence in that of T'), because then P is also a T-path and T is normal in G by assumption. If not, then y is one end of P, so P lies in C except for its other end z, which lies in $N(C)$. Then $z \leqslant x$, by the choice of x. For our proof that y and z are comparable it thus suffices to show that $x < y$, i.e. that $x \in rT'y$. This, however, is clear since y is a leaf of T' with neighbour x. □

1.6 Bipartite graphs

r-partite Let $r \geqslant 2$ be an integer. A graph $G = (V, E)$ is called *r-partite* if V admits a partition into r classes such that every edge has its ends in different classes: vertices in the same partition class must not be adjacent. Instead of '2-partite' one usually says *bipartite*. *bipartite*

complete r-partite An r-partite graph in which *every* two vertices from different partition classes are adjacent is called *complete*; the complete r-partite graphs for all r together are the *complete multipartite* graphs. The

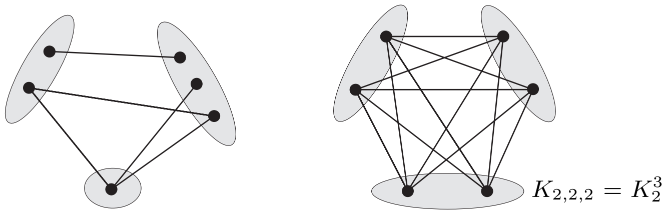

Fig. 1.6.1. Two 3-partite graphs

complete r-partite graph $\overline{K^{n_1}} * \ldots * \overline{K^{n_r}}$ is denoted by $K_{n_1,\ldots,n_r}$; if $n_1 = \ldots = n_r =: s$, we abbreviate this to K_s^r. Thus, K_s^r is the complete r-partite graph in which every partition class contains exactly s vertices.[5] (Figure 1.6.1 shows the example of the octahedron K_2^3; compare its drawing with that in Figure 1.4.3.) Graphs of the form $K_{1,n}$ are called *stars*.

$K_{n_1,\ldots,n_r}$ K_s^r *star*

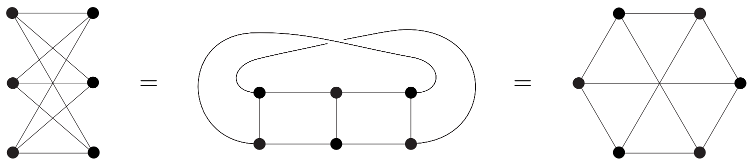

Fig. 1.6.2. Three drawings of the bipartite graph $K_{3,3} = K_3^2$

Clearly, a bipartite graph cannot contain an *odd cycle*, a cycle of odd length. In fact, the bipartite graphs are characterized by this property: *odd cycle*

Proposition 1.6.1. *A graph is bipartite if and only if it contains no odd cycle.* [5.3.1] [6.4.2]

Proof. Let $G = (V, E)$ be a graph without odd cycles; we show that G is bipartite. Clearly a graph is bipartite if all its components are bipartite or trivial, so we may assume that G is connected. Let T be a spanning tree in G, pick a root $r \in T$, and denote the associated tree-order on V by $\leqslant_T$. For each $v \in V$, the unique path rTv has odd or even length. This defines a bipartition of V; we show that G is bipartite with this partition. (1.5.1)

Let $e = xy$ be an edge of G. If $e \in T$, with $x <_T y$ say, then $rTy = rTxy$ and so x and y lie in different partition classes. If $e \notin T$ then $C_e := xTy + e$ is a cycle (Fig. 1.6.3), and by the case treated already the vertices along xTy alternate between the two classes. Since C_e is even by assumption, x and y again lie in different classes. □

[5] Note that we obtain a K_s^r if we replace each vertex of a K^r by an independent s-set; our notation of K_s^r is intended to hint at this connection.

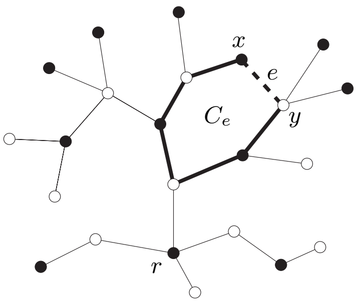

Fig. 1.6.3. The cycle C_e in $T + e$

1.7 Contraction and minors

In Section 1.1 we saw two fundamental containment relations between graphs: the subgraph relation, and the 'induced subgraph' relation. In this section we meet another: the minor relation.

G/e
contraction
v_e

Let $e = xy$ be an edge of a graph $G = (V, E)$. By G/e we denote the graph obtained from G by *contracting* the edge e into a new vertex v_e, which becomes adjacent to all the former neighbours of x and of y. Formally, G/e is a graph (V', E') with vertex set $V' := (V \setminus \{x, y\}) \cup \{v_e\}$ (where v_e is the 'new' vertex, i.e. $v_e \notin V \cup E$) and edge set

$$E' := \Big\{ vw \in E \mid \{v, w\} \cap \{x, y\} = \emptyset \Big\} \cup \Big\{ v_e w \mid xw \in E \setminus \{e\} \text{ or } yw \in E \setminus \{e\} \Big\}.$$

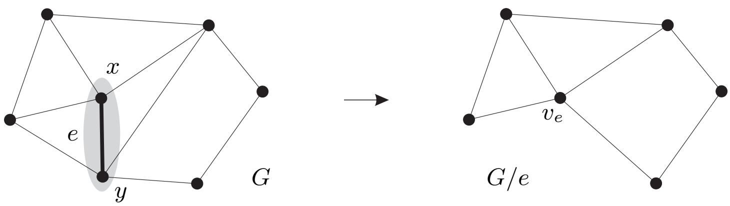

Fig. 1.7.1. Contracting the edge $e = xy$

MX
branch sets

More generally, if X is another graph and $\{V_x \mid x \in V(X)\}$ is a partition of V into connected subsets such that, for any two vertices $x, y \in X$, there is a V_x–V_y edge in G if and only if $xy \in E(X)$, we call G an MX and write[6] $G = MX$ (Fig. 1.7.2). The sets V_x are the *branch sets* of this MX. Intuitively, we obtain X from G by contracting every

6 Thus formally, the expression MX—where M stands for 'minor'; see below—refers to a whole class of graphs, and $G = MX$ means (with slight abuse of notation) that G belongs to this class.

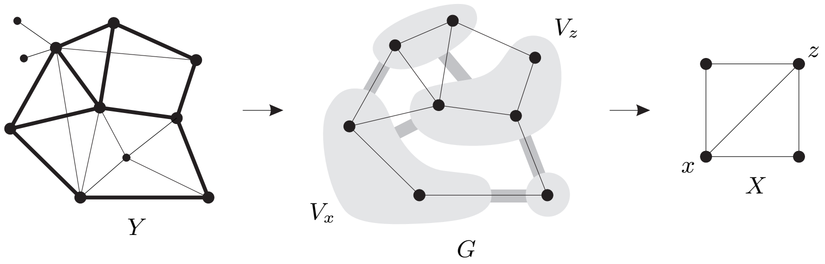

Fig. 1.7.2. $Y \supseteq G = MX$, so X is a minor of Y

branch set to a single vertex and deleting any 'parallel edges' or 'loops' that may arise.

If $V_x = U \subseteq V$ is one of the branch sets above and every other branch set consists just of a single vertex, we also write G/U for the graph X and v_U for the vertex $x \in X$ to which U contracts, and think of the rest of X as an induced subgraph of G. The contraction of a single edge uu' defined earlier can then be viewed as the special case of $U = \{ u, u' \}$. *G/U* *v_U*

Proposition 1.7.1. *G is an MX if and only if X can be obtained from G by a series of edge contractions, i.e. if and only if there are graphs $G_0, \dots, G_n$ and edges $e_i \in G_i$ such that $G_0 = G$, $G_n \simeq X$, and $G_{i+1} = G_i/e_i$ for all $i < n$.*

Proof. Induction on $|G| - |X|$. □

If $G = MX$ is a subgraph of another graph Y, we call X a *minor* of Y and write $X \preccurlyeq Y$. Note that every subgraph of a graph is also its minor; in particular, every graph is its own minor. By Proposition 1.7.1, any minor of a graph can be obtained from it by first deleting some vertices and edges, and then contracting some further edges. Conversely, any graph obtained from another by repeated deletions and contractions (in any order) is its minor: this is clear for one deletion or contraction, and follows for several from the transitivity of the minor relation (Proposition 1.7.3). *minor; $\preccurlyeq$*

If we replace the edges of X with independent paths between their ends (so that none of these paths has an inner vertex on another path or in X), we call the graph G obtained a *subdivision* of X and write $G = TX$.[7] If $G = TX$ is the subgraph of another graph Y, then X is a *topological minor* of Y (Fig. 1.7.3). *subdivision* *TX* *topological minor*

[7] So again TX denotes an entire class of graphs: all those which, viewed as a topological space in the obvious way, are homeomorphic to X. The T in TX stands for 'topological'.

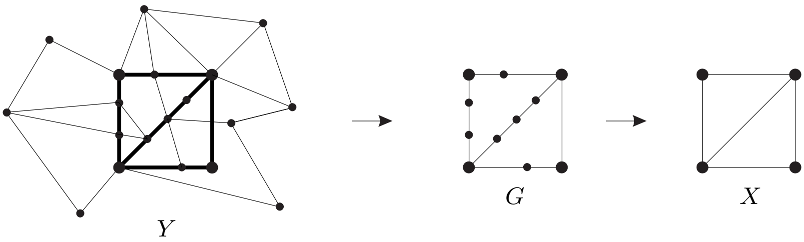

Fig. 1.7.3. $Y \supseteq G = TX$, so X is a topological minor of Y

branch vertices

If $G = TX$, we view $V(X)$ as a subset of $V(G)$ and call these vertices the *branch vertices* of G; the other vertices of G are its *subdividing vertices*. Thus, all subdividing vertices have degree 2, while the branch vertices retain their degree from X.

[4.4.2] [8.3.1]

Proposition 1.7.2.

(i) *Every TX is also an MX (Fig. 1.7.4); thus, every topological minor of a graph is also its (ordinary) minor.*

(ii) *If $\Delta(X) \leqslant 3$, then every MX contains a TX; thus, every minor with maximum degree at most 3 of a graph is also its topological minor.* □

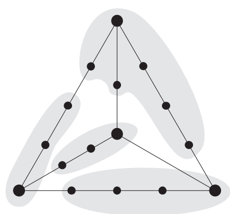

Fig. 1.7.4. A subdivision of K^4 viewed as an MK^4

[12.4.1] **Proposition 1.7.3.** *The minor relation $\preccurlyeq$ and the topological-minor relation are partial orderings on the class of finite graphs, i.e. they are reflexive, antisymmetric and transitive.* □

1.8 Euler tours

Any mathematician who happens to find himself in the East Prussian city of Königsberg (and in the 18th century) will lose no time to follow the great Leonhard Euler's example and inquire about a round trip through

Fig. 1.8.1. The bridges of Königsberg (anno 1736)

the old city that traverses each of the bridges shown in Figure 1.8.1 exactly once.

Thus inspired,[8] let us call a closed walk in a graph an *Euler tour* if it traverses every edge of the graph exactly once. A graph is *Eulerian* if it admits an Euler tour. *Eulerian*

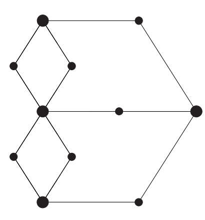

Fig. 1.8.2. A graph formalizing the bridge problem

Theorem 1.8.1. (Euler 1736)
A connected graph is Eulerian if and only if every vertex has even degree. [2.1.5] [10.3.3]

Proof. The degree condition is clearly necessary: a vertex appearing k times in an Euler tour (or $k+1$ times, if it is the starting and finishing vertex and as such counted twice) must have degree $2k$.

[8] Anyone to whom such inspiration seems far-fetched, even after contemplating Figure 1.8.2, may seek consolation in the *multigraph* of Figure 1.10.1.

Conversely, let G be a connected graph with all degrees even, and let

$$W = v_0 e_0 \dots e_{\ell-1} v_\ell$$

be a longest walk in G using no edge more than once. Since W cannot be extended, it already contains all the edges at v_ℓ. By assumption, the number of such edges is even. Hence $v_\ell = v_0$, so W is a closed walk.

Suppose W is not an Euler tour. Then G has an edge e outside W but incident with a vertex of W, say $e = uv_i$. (Here we use the connectedness of G, as in the proof of Proposition 1.4.1.) Then the walk

$$uev_i e_i \dots e_{\ell-1} v_\ell e_0 \dots e_{i-1} v_i$$

is longer than W, a contradiction. □

1.9 Some linear algebra

vertex space $\mathcal{V}(G)$

Let $G = (V, E)$ be a graph with n vertices and m edges, say $V = \{ v_1, \dots, v_n \}$ and $E = \{ e_1, \dots, e_m \}$. The *vertex space* $\mathcal{V}(G)$ of G is the vector space over the 2-element field $\mathbb{F}_2 = \{ 0, 1 \}$ of all functions $V \to \mathbb{F}_2$. Every element of $\mathcal{V}(G)$ corresponds naturally to a subset of V, the set of those vertices to which it assigns a 1, and every subset of V is uniquely represented in $\mathcal{V}(G)$ by its indicator function. We may thus think of $\mathcal{V}(G)$ as the power set of V made into a vector space: the sum $U + U'$ of two vertex sets $U, U' \subseteq V$ is their symmetric difference (why?), and $U = -U$ for all $U \subseteq V$. The zero in $\mathcal{V}(G)$, viewed in this way, is the empty (vertex) set $\emptyset$. Since $\{ \{ v_1 \}, \dots, \{ v_n \} \}$ is a basis of $\mathcal{V}(G)$, its *standard basis*, we have $\dim \mathcal{V}(G) = n$.

$+$

edge space $\mathcal{E}(G)$

standard basis

In the same way as above, the functions $E \to \mathbb{F}_2$ form the *edge space* $\mathcal{E}(G)$ of G: its elements are the subsets of E, vector addition amounts to symmetric difference, $\emptyset \subseteq E$ is the zero, and $F = -F$ for all $F \subseteq E$. As before, $\{ \{ e_1 \}, \dots, \{ e_m \} \}$ is the *standard basis* of $\mathcal{E}(G)$, and $\dim \mathcal{E}(G) = m$.

Since the edges of a graph carry its essential structure, we shall mostly be concerned with the edge space. Given two edge sets $F, F' \in \mathcal{E}(G)$ and their coefficients $\lambda_1, \dots, \lambda_m$ and $\lambda'_1, \dots, \lambda'_m$ with respect to the standard basis, we write

$\langle F, F' \rangle$

$$\langle F, F' \rangle := \lambda_1 \lambda'_1 + \dots + \lambda_m \lambda'_m \in \mathbb{F}_2 .$$

Note that $\langle F, F' \rangle = 0$ may hold even when $F = F' \neq \emptyset$: indeed, $\langle F, F' \rangle = 0$ if and only if F and F' have an even number of edges in common. Given a subspace $\mathcal{F}$ of $\mathcal{E}(G)$, we write

$$\mathcal{F}^\perp := \{ D \in \mathcal{E}(G) \mid \langle F, D \rangle = 0 \text{ for all } F \in \mathcal{F} \} .$$ $\mathcal{F}^\perp$

This is again a subspace of $\mathcal{E}(G)$ (the space of all vectors solving a certain set of linear equations—which?), and we have

$$\dim \mathcal{F} + \dim \mathcal{F}^\perp = m .$$

The *cycle space* $\mathcal{C} = \mathcal{C}(G)$ is the subspace of $\mathcal{E}(G)$ spanned by all the cycles in G—more precisely, by their edge sets.[9] The dimension of $\mathcal{C}(G)$ is the *cyclomatic number* of G. *cycle space* $\mathcal{C}(G)$

Proposition 1.9.1. *The induced cycles in G generate its entire cycle space.* [3.2.3]

Proof. By definition of $\mathcal{C}(G)$ it suffices to show that the induced cycles in G generate every cycle $C \subseteq G$ with a chord e. This follows at once by induction on $|C|$: the two cycles in $C + e$ with e but no other edge in common are shorter than C, and their symmetric difference is precisely C. □

Proposition 1.9.2. *An edge set $F \subseteq E$ lies in $\mathcal{C}(G)$ if and only if every vertex of (V, F) has even degree.* [4.5.1]

Proof. The forward implication holds by induction on the number of cycles needed to generate F, the backward implication by induction on the number of cycles in (V, F). □

If $\{ V_1, V_2 \}$ is a partition of V, the set $E(V_1, V_2)$ of all the edges of G *crossing* this partition is called a *cut*. Recall that for $V_1 = \{ v \}$ this cut is denoted by $E(v)$. *cut*

Proposition 1.9.3. *Together with $\emptyset$, the cuts in G form a subspace $\mathcal{C}^*$ of $\mathcal{E}(G)$. This space is generated by cuts of the form $E(v)$.* [4.6.3]

Proof. Let $\mathcal{C}^*$ denote the set of all cuts in G, together with $\emptyset$. To prove that $\mathcal{C}^*$ is a subspace, we show that for all $D, D' \in \mathcal{C}^*$ also $D + D'$ $(= D - D')$ lies in $\mathcal{C}^*$. Since $D + D = \emptyset \in \mathcal{C}^*$ and $D + \emptyset = D \in \mathcal{C}^*$, we may assume that D and D' are distinct and non-empty. Let $\{ V_1, V_2 \}$ and $\{ V_1', V_2' \}$ be the corresponding partitions of V. Then $D + D'$ consists of all the edges that cross one of these partitions but not the other (Fig. 1.9.1). But these are precisely the edges between $(V_1 \cap V_1') \cup (V_2 \cap V_2')$ and $(V_1 \cap V_2') \cup (V_2 \cap V_1')$, and by $D \neq D'$ these two

[9] For simplicity, we shall not normally distinguish between cycles and their edge sets in connection with the cycle space.

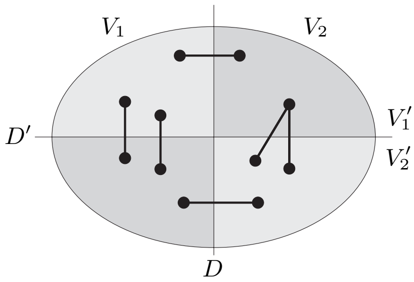

Fig. 1.9.1. Cut edges in $D + D'$

sets form another partition of V. Hence $D + D' \in \mathcal{C}^*$, and $\mathcal{C}^*$ is indeed a subspace of $\mathcal{E}(G)$.

Our second assertion, that the cuts $E(v)$ generate all of $\mathcal{C}^*$, follows from the fact that every edge $xy \in G$ lies in exactly two such cuts (in $E(x)$ and in $E(y)$); thus every partition $\{ V_1, V_2 \}$ of V satisfies $E(V_1, V_2) = \sum_{v \in V_1} E(v)$. □

cut space $\mathcal{C}^*(G)$

The subspace $\mathcal{C}^* =: \mathcal{C}^*(G)$ of $\mathcal{E}(G)$ from Proposition 1.9.3 will be called the *cut space* of G. It is not difficult to find among the cuts $E(v)$ an explicit basis for $\mathcal{C}^*(G)$, and thus to determine its dimension (exercise); together with Theorem 1.9.5 this yields an independent proof of Theorem 1.9.6.

The following lemma will be useful when we study the duality of plane graphs in Chapter 4.6:

[4.6.2] **Lemma 1.9.4.** *The minimal cuts in a connected graph generate its entire cut space.*

Proof. Note first that a cut in a connected graph $G = (V, E)$ is minimal if and only if both sets in the corresponding partition of V are connected in G. Now consider any connected subgraph $C \subseteq G$. If D is a component of $G - C$, then also $G - D$ is connected (Fig. 1.9.2); the edges between D

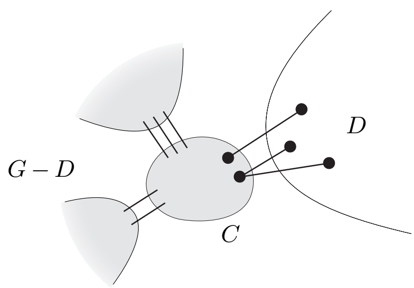

Fig. 1.9.2. $G - D$ is connected, and $E(C, D)$ a minimal cut

and $G - D$ thus form a minimal cut. By choice of D, this cut is precisely the set $E(C, D)$ of all C–D edges in G.

To prove the lemma, let a partition $\{ V_1, V_2 \}$ of V be given, and consider a component C of $G[V_1]$. Then $E(C, V_2) = E(C, G - C)$ is the disjoint union of the edge sets $E(C, D)$ over all components D of $G - C$, and is thus the disjoint union of minimal cuts (see above). Now the disjoint union of all these edge sets $E(C, V_2)$, taken over all the components C of $G[V_1]$, is precisely our cut $E(V_1, V_2)$. So this cut is generated by minimal cuts, as claimed. □

Theorem 1.9.5. *The cycle space $\mathcal{C}$ and the cut space $\mathcal{C}^*$ of any graph satisfy*

$$\mathcal{C} = \mathcal{C}^{*\perp} \quad \textit{and} \quad \mathcal{C}^* = \mathcal{C}^{\perp}.$$

Proof. Let us consider a graph $G = (V, E)$. Clearly, any cycle in G has an even number of edges in each cut. This implies $\mathcal{C} \subseteq \mathcal{C}^{*\perp}$.

Conversely, recall from Proposition 1.9.2 that for every edge set $F \notin \mathcal{C}$ there exists a vertex v incident with an odd number of edges in F. Then $\langle E(v), F \rangle = 1$, so $E(v) \in \mathcal{C}^*$ implies $F \notin \mathcal{C}^{*\perp}$. This completes the proof of $\mathcal{C} = \mathcal{C}^{*\perp}$.

To prove $\mathcal{C}^* = \mathcal{C}^{\perp}$, it now suffices to show $\mathcal{C}^* = (\mathcal{C}^{*\perp})^{\perp}$. Here $\mathcal{C}^* \subseteq (\mathcal{C}^{*\perp})^{\perp}$ follows directly from the definition of $\perp$. But since

$$\dim \mathcal{C}^* + \dim \mathcal{C}^{*\perp} = m = \dim \mathcal{C}^{*\perp} + \dim (\mathcal{C}^{*\perp})^{\perp},$$

$\mathcal{C}^*$ has the same dimension as $(\mathcal{C}^{*\perp})^{\perp}$, so $\mathcal{C}^* = (\mathcal{C}^{*\perp})^{\perp}$ as claimed. □

Theorem 1.9.6. *Every connected graph G with n vertices and m edges satisfies* [4.5.1]

$$\dim \mathcal{C}(G) = m - n + 1 \quad \textit{and} \quad \dim \mathcal{C}^*(G) = n - 1.$$

Proof. Let $G = (V, E)$. As $\dim \mathcal{C} + \dim \mathcal{C}^* = m$ by Theorem 1.9.5, it suffices to find $m - n + 1$ linearly independent vectors in $\mathcal{C}$ and $n - 1$ linearly independent vectors in $\mathcal{C}^*$: since these numbers add up to m, neither the dimension of $\mathcal{C}$ nor that of $\mathcal{C}^*$ can then be strictly greater. (1.5.1) (1.5.3)

Let T be a spanning tree in G. By Corollary 1.5.3, T has $n - 1$ edges, so $m - n + 1$ edges of G lie outside T. For each of these $m - n + 1$ edges $e \in E \setminus E(T)$, the graph $T + e$ contains a cycle C_e (see Fig. 1.6.3 and Theorem 1.5.1 (iv)). Since none of the edges e lies on $C_{e'}$ for $e' \neq e$, these $m - n + 1$ cycles are linearly independent.

For each of the $n - 1$ edges $e \in T$, the graph $T - e$ has exactly two components (Theorem 1.5.1 (iii)), and the set D_e of edges in G between these components form a cut (Fig.1.9.3). Since none of the edges $e \in T$ lies in $D_{e'}$ for $e' \neq e$, these $n - 1$ cuts are linearly independent. □

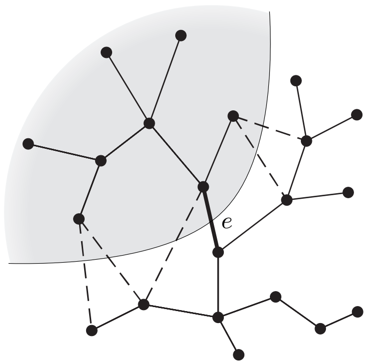

Fig. 1.9.3. The cut D_e

incidence matrix

The *incidence matrix* $B = (b_{ij})_{n \times m}$ of a graph $G = (V, E)$ with $V = \{ v_1, \dots, v_n \}$ and $E = \{ e_1, \dots, e_m \}$ is defined over $\mathbb{F}_2$ by

$$b_{ij} := \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise.} \end{cases}$$

As usual, let B^t denote the transpose of B. Then B and B^t define linear maps $B\colon \mathcal{E}(G) \to \mathcal{V}(G)$ and $B^t\colon \mathcal{V}(G) \to \mathcal{E}(G)$ with respect to the standard bases.

Proposition 1.9.7.

(i) *The kernel of B is $\mathcal{C}(G)$.*

(ii) *The image of B^t is $\mathcal{C}^*(G)$.* □

adjacency matrix

The *adjacency matrix* $A = (a_{ij})_{n \times n}$ of G is defined by

$$a_{ij} := \begin{cases} 1 & \text{if } v_i v_j \in E \\ 0 & \text{otherwise.} \end{cases}$$

Our last proposition establishes a simple connection between A and B (now viewed as real matrices). Let D denote the real diagonal matrix $(d_{ij})_{n \times n}$ with $d_{ii} = d(v_i)$ and $d_{ij} = 0$ otherwise.

Proposition 1.9.8. $BB^t = A + D$. □