Reinhard Diestel

Graph Theory

5th Electronic Edition 2016

This is the 5th eBook edition of the above Springer book, from their series *Graduate Texts in Mathematics*, vol. 173.

There is also a platform-independent Professional Edition, which can be annotated, printed, and shared over many devices. The Professional Edition also includes a section with hints for the exercises. See

`http://diestel-graph-theory.com/`

where also reviews and any errata are posted.

For iPad users there is a free iOS app for the book, which can load both the Professional Edition and the eBook, either whole or by chapter.

Lecturers basing their course on the book are entitled to a free personal Professional Edition as part of a *course notes licence*, which allows them to distribute unlimited handouts in class. See here for details.

To Dagmar

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 $\square$. 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 $\square$.

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 12.4.3, 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*

About the third edition

There is no denying that this book has grown. Is it still as 'lean and concentrating on the essential' as I said it should be when I wrote the preface to the first edition, now almost eight years ago?

I believe that it is, perhaps now more than ever. So why the increase in volume? Part of the answer is that I have continued to pursue the original dual aim of offering two different things between one pair of covers:

- a reliable first introduction to graph theory that can be used either for personal study or as a course text;
- a graduate text that also offers some depth on the most important topics.

For each of these aims, some material has been added. Some of this covers new topics, which can be included or skipped as desired. An example at the introductory level is the new section on packing and covering with the Erdős-Pósa theorem, or the inclusion of the stable marriage theorem in the matching chapter. An example at the graduate level is the Robertson-Seymour structure theorem for graphs without a given minor: a result that takes a few lines to state, but one which is increasingly relied on in the literature, so that an easily accessible reference seems desirable. Another addition, also in the chapter on graph minors, is a new proof of the 'Kuratowski theorem for higher surfaces'—a proof which illustrates the interplay between graph minor theory and surface topology better than was previously possible. The proof is complemented by an appendix on surfaces, which supplies the required background and also sheds some more light on the proof of the graph minor theorem.

Changes that affect previously existing material are rare, except for countless local improvements intended to consolidate and polish rather than change. I am aware that, as this book is increasingly adopted as a course text, there is a certain desire for stability. Many of these local improvements are the result of generous feedback I got from colleagues using the book in this way, and I am very grateful for their help and advice.

There are also some local additions. Most of these developed from my own notes, pencilled in the margin as I prepared to teach from the book. They typically complement an important but technical proof, when I felt that its essential ideas might get overlooked in the formal write-up. For example, the proof of the Erdős-Stone theorem now has an informal post-mortem that looks at how exactly the regularity lemma comes to be applied in it. Unlike the formal proof, the discussion starts out from the main idea, and finally arrives at how the parameters to be declared at the start of the formal proof must be specified. Similarly, there is now a discussion pointing to some ideas in the proof of the perfect graph theorem. However, in all these cases the formal proofs have been left essentially untouched.

The only substantial change to existing material is that the old Theorem 8.1.1 (that cr^2n edges force a TK^r) seems to have lost its nice (and long) proof. Previously, this proof had served as a welcome opportunity to explain some methods in sparse extremal graph theory. These methods have migrated to the connectivity chapter, where they now live under the roof of the new proof by Thomas and Wollan that $8kn$ edges make a $2k$-connected graph k-linked. So they are still there, leaner than ever before, and just presenting themselves under a new guise. As a consequence of this change, the two earlier chapters on dense and sparse extremal graph theory could be reunited, to form a new chapter appropriately named as *Extremal Graph Theory*.

Finally, there is an entirely new chapter, on infinite graphs. When graph theory first emerged as a mathematical discipline, finite and infinite graphs were usually treated on a par. This has changed in recent years, which I see as a regrettable loss: infinite graphs continue to provide a natural and frequently used bridge to other fields of mathematics, and they hold some special fascination of their own. One aspect of this is that proofs often have to be more constructive and algorithmic in nature than their finite counterparts. The infinite version of Menger's theorem in Section 8.4 is a typical example: it offers algorithmic insights into connectivity problems in networks that are invisible to the slick inductive proofs of the finite theorem given in Chapter 3.3.

Once more, my thanks go to all the readers and colleagues whose comments helped to improve the book. I am particularly grateful to Imre Leader for his judicious comments on the whole of the infinite chapter; to my graph theory seminar, in particular to Lilian Matthiesen and Philipp Sprüssel, for giving the chapter a test run and solving all its exercises (of which eighty survived their scrutiny); to Agelos Georgakopoulos for much proofreading elsewhere; to Melanie Win Myint for recompiling the index and extending it substantially; and to Tim Stelldinger for nursing the whale on page 404 until it was strong enough to carry its baby dinosaur.

May 2005 *RD*

About the fourth edition

In this fourth edition there are few substantial additions of new material, but many improvements.

As with previous new editions, there are countless small and subtle changes to further elucidate a particular argument or concept. When prompted by reader feedback, for which I am always grateful, I still try to recast details that have been found harder than they should be. These can be very basic; a nice example, this time, is the definition of a minor in Chapter 1.

At a more substantial level, there are several new and simpler proofs of classical results, in one case reducing the already shortened earlier proof to half its length (and twice its beauty). These newly added proofs include the marriage theorem, the tree packing theorem, Tutte's cycle space and wheel theorem, Fleischner's theorem on Hamilton cycles, and the threshold theorem for the edge probability guaranteeing a specified type of subgraph. There are also one or two genuinely new theorems. One of these is an ingenious local degree condition for the existence of a Hamilton cycle, due to Asratian and Khachatrian, that implies a number of classical hamiltonicity theorems.

In some sections I have reorganized the material slightly, or rewritten the narrative. Typically, these are sections that had grown over the previous three editions, and this was beginning to affect their balance of material and momentum. As the book remains committed to offering not just a collection of theorems and proofs, but tries whenever possible to indicate a somewhat larger picture in which these have their place, maintaining its original freshness and flow remains a challenge that I enjoy trying to meet.

Finally, the book has its own dedicated website now, at

`http://diestel-graph-theory.com/`

Potentially, this offers opportunities for more features surrounding the book than the traditional free online edition and a dwindling collection of misprints. If you have any ideas and would like to see them implemented, do let me know.

May 2010 *RD*

About the fifth edition

This fifth edition of the book is again a major overhaul, in the spirit of its first and third edition.

I have rewritten Chapter 12 on graph minors to take account of recent developments. In addition to many smaller updates it offers a new proof of the tree-width duality theorem, due to Mazoit, which has not otherwise been published. More fundamentally, I have added a section on tangles. Originally devised by Robertson and Seymour as a technical device for their proof of the graph minor theorem, tangles have turned out to be much more fundamental than this: they define a new paradigm for identifying highly connected parts in a graph. Unlike earlier attempts at defining such substructures—in terms of, say, highly connected subgraphs, minors, or topological minors—tangles do not attempt to pin down this substructure in terms of vertices, edges, or connecting paths, but seek to capture it indirectly by orienting all the low-order separations of the graph towards it. In short, we no longer ask *what* exactly the highly connected region is, but only *where* it is. For many applications, this is exactly what matters. Moreover, this more abstract notion of high local connectivity can easily be transported to contexts outside graph theory. This, in turn, makes graph minor theory applicable beyond graph theory itself in a new way, via tangles. I have written the new section on tangles from this modern perspective.

Chapter 2 has a newly written section on tree packing and covering. I rewrote it from scratch to take advantage of a beautiful new unified theorem containing both aspects at once: the *packing-covering theorem* of Bowler and Carmesin. While their original result was proved for matroids, its graph version has a very short and self-contained proof. This proof is given in Chapter 2.4, and again is not found in print elsewhere.

Chapter 8, on infinite graphs, now treats the topological aspects of locally finite graphs more thoroughly. It puts the Freudenthal compactification of a graph G into perspective by describing it, in addition, as an inverse limit of the finite contraction minors of G. Readers with a background in group theory will find this familiar.

As always, there are countless small improvements to the narrative, proofs, and exercises. My thanks go to all those who suggested these.

Finally, I have made two adjustments to help ensure that the exercises remain usable in class at a time of instant internet access. The Hints appendix still exists, but has been relegated to the professional electronic edition so that lecturers can decide which hints to give and which not. Similarly, exercises asking for a proof of a named theorem no longer mention this name, so that the proof cannot simply be searched for. However if you know the name and wish to find the exercise, the index still has a name entry that will take you to the right page.

July 2016 *RD*

Contents

* Sections marked by an asterisk are recommended for a first course.
Of sections marked (*), the beginning is recommended for a first course.

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. For deviations for multigraphs see Section 1.10.

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 $\overline{i} := i + n\mathbb{Z}$. When we regard $\mathbb{Z}_2 = \{\overline{0}, \overline{1}\}$ as a field, we also denote it as $\mathbb{F}_2 = \{0, 1\}$. 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 taken at base 2; the natural logarithm will be denoted by 'ln'. The expressions $x := y$ and $y =: x$ mean that x is being defined as y. $\mathbb{Z}_n$ $\lfloor x \rfloor, \lceil x \rceil$ log, ln

A set $\mathcal{A} = \{A_1, \ldots, A_k\}$ of disjoint subsets of a set A is a *partition* of A if the union $\bigcup \mathcal{A}$ of all the sets $A_i \in \mathcal{A}$ is A and $A_i \neq \emptyset$ for every i. Another partition $\{A'_1, \ldots, A'_\ell\}$ of A *refines* the partition $\mathcal{A}$ if each A'_i is contained in some A_j. By $[A]^k$ we 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*. *partition* $\bigcup \mathcal{A}$ $[A]^k$ *k-set*

1.1 Graphs

graph *vertex* *edge*

A *graph* is a pair $G = (V, E)$ of sets such that $E \subseteq [V]^2$; 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 of which pairs of vertices form an edge and which do not.

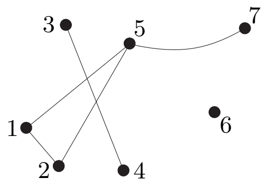


Fig. 1.1.1. The graph on $V = \{1, \ldots, 7\}$ with edge set $E = \{\{1,2\},\{1,5\},\{2,5\},\{3,4\},\{5,7\}\}$

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*, *infinite*, *countable* and so on according to their order. Except in Chapter 8, our graphs will be finite unless otherwise stated.

$\emptyset$ *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 $\{x, y\}$ 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* if no two of its elements are adjacent. Independent sets of vertices are also called *stable sets*.

independent

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. A map $\varphi \colon V \to V'$ is a *homomorphism* from G to G' if it preserves the adjacency of vertices, that is, if $\{\varphi(x), \varphi(y)\} \in E'$ whenever $\{x, y\} \in E$. Then, in particular, for every vertex x' in the image of φ its inverse image $\varphi^{-1}(x')$ is an independent set of vertices in G. If φ is bijective and its inverse φ^{-1} is also a homomorphism (so that $xy \in E \Leftrightarrow \varphi(x)\varphi(y) \in E'$ for all $x, y \in V$), we call φ an *isomorphism*, say that G and G' are *isomorphic*, and write $G \simeq G'$. An isomorphism from G to itself is an *automorphism* of G.

homomorphism *isomorphic* $\simeq$

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. If we wish to emphasize that we are only interested in the isomorphism type of a given graph, we informally refer to it as an *abstract graph*.

$=$

A class of graphs that is closed under isomorphism is called a *graph property*. For example, 'containing a triangle' is a graph property: if G contains three pairwise adjacent vertices then so does every graph isomorphic to G. 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.

property *invariant*

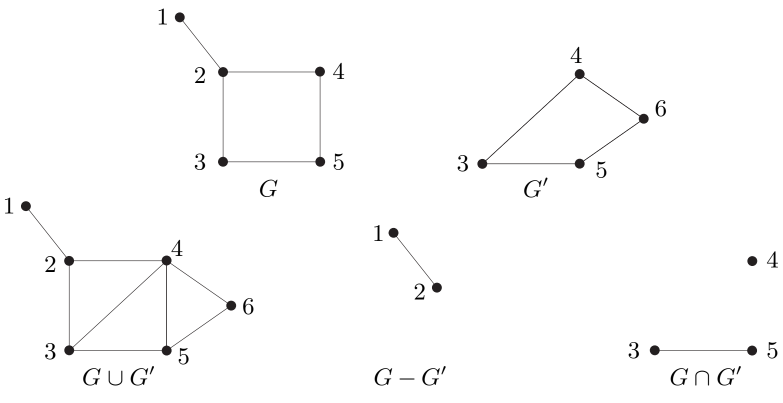


Fig. 1.1.2. Union, difference and intersection; the vertices 2,3,4 induce (or span) a triangle in $G \cup G'$ but not in G

We set $G \cup G' := (V \cup V', E \cup E')$ and $G \cap G' := (V \cap V', E \cap E')$.

$G \cap G'$

subgraph
$G' \subseteq G$

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'. If $G' \subseteq G$ and $G' \neq G$, then G' is a *proper subgraph* of G.

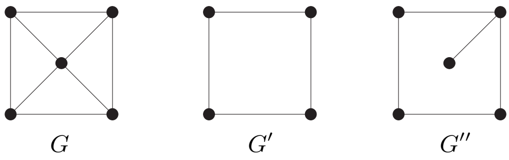


Fig. 1.1.3. A graph G with subgraphs G' and G'': G' is an induced subgraph of G, but G'' is not

induced subgraph
$G[U]$
spanning

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 $V' = V$.

$-$
$+$
edge-maximal

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 (V, F) with $F \supsetneq E$ does.

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.

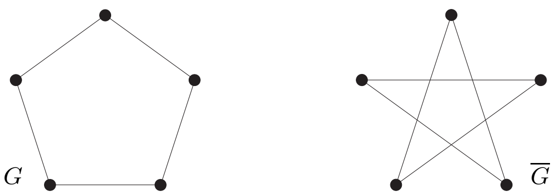


Fig. 1.1.4. A graph isomorphic to its complement

1.2 The degree of a vertex

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

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.* [10.3.1]

Proof. As $|E| = \frac{1}{2}\sum_{v \in V} d(v)$ is an integer, $\sum_{v \in V} d(v)$ is even. $\square$

---

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

[2] but not for multigraphs; see Section 1.10

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:

[1.4.3] [7.2.2] **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. $\square$

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 *endvertices* or *ends*; the vertices $x_1, \ldots, x_{k-1}$ are the *inner* vertices of P. 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, $P^0 = K^1$.

length *P^k*

We often refer to a path by the natural sequence of its vertices,[3]

[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.

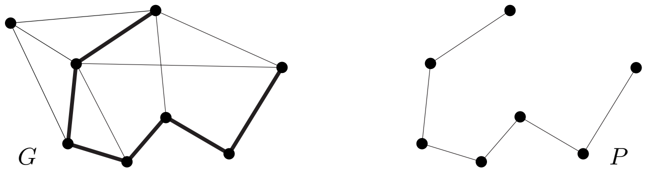


Fig. 1.3.1. A path $P = P^6$ in G

writing, say, $P = x_0 x_1 \ldots x_k$ and calling P a path *from* x_0 *to* x_k (as well as *between* x_0 and x_k).

For $0 \leqslant i \leqslant j \leqslant k$ we write $xPy,\ \mathring{P}$

$$\begin{aligned} Px_i &:= x_0 \ldots x_i \\ x_i P &:= x_i \ldots x_k \\ x_i P x_j &:= x_i \ldots x_j \end{aligned}$$

and

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

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 $PxQyR$. $PxQyR$

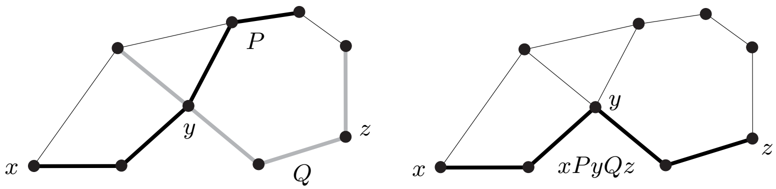


Fig. 1.3.2. Paths P, Q and $xPyQz$

Given sets A, B of vertices, we call $P = x_0 \ldots x_k$ 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. *A–B path* *independent*

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. *H-path*

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.

length
C^k

girth $g(G)$
circum-ference
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).

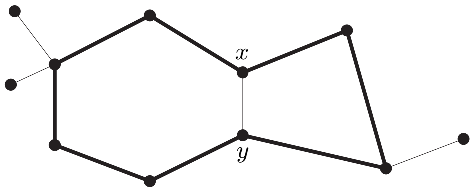


Fig. 1.3.3. A cycle C^8 with chord xy, and induced cycles C^6, C^4

If a graph has large minimum degree, it contains long paths and cycles (see also Exercise 9):

[1.4.3]
[7.2.2]

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 $\delta(G)+1$. $\square$

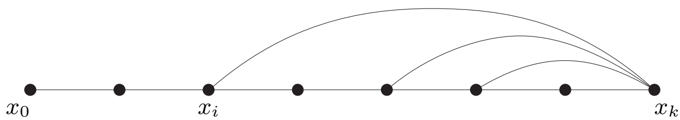


Fig. 1.3.4. A longest path $x_0 \dots x_k$, and the neighbours of x_k

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
$d(x,y)$

diameter
$\mathrm{diam}(G)$

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:

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

Proof. Let C be a shortest cycle in G. If $g(G) \geqslant 2\operatorname{diam}(G) + 2$, then C has two vertices whose distance in C is at least $\operatorname{diam}(G) + 1$. 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 $\operatorname{rad}(G)$. Thus, formally, $\operatorname{rad}(G) = \min_{x \in V(G)} \max_{y \in V(G)} d_G(x,y)$. As one easily checks (exercise), we have

central

radius
$\operatorname{rad}(G)$

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

Diameter and radius are not related to minimum, average or maximum degree if we say nothing about the order of the graph. However, graphs of large diameter and minimum degree must be large (larger than forced by each of the two parameters alone; see Exercise 10), and graphs of small diameter and maximum degree must be small:

[9.4.1]
[9.4.2]

Proposition 1.3.3. *A graph G of radius at most k and maximum degree at most $d \geqslant 3$ has fewer than $\frac{d}{d-2}(d-1)^k$ vertices.*

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 $V(G) = \bigcup_{i=0}^{k} D_i$. Clearly $|D_0| = 1$ and $|D_1| \leqslant d$. For $i \geqslant 1$ we have $|D_{i+1}| \leqslant (d-1)|D_i|$, because every vertex in D_{i+1} is a neighbour of a vertex in D_i (why?), and each vertex in D_i has at most $d-1$ neighbours in D_{i+1} (since it has another neighbour in D_{i-1}). Thus $|D_{i+1}| \leqslant d(d-1)^i$ for all $i < k$ by induction, giving

$$|G| \leqslant 1 + d\sum_{i=0}^{k-1}(d-1)^i = 1 + \frac{d}{d-2}\big((d-1)^k - 1\big) < \frac{d}{d-2}(d-1)^k.$$

□

Similarly, we can bound the order of G from below by assuming that both its minimum degree and girth are large. For $d \in \mathbb{R}$ and $g \in \mathbb{N}$ let

$$n_0(d,g) := \begin{cases} 1 + d\sum_{i=0}^{r-1}(d-1)^i & \text{if } g =: 2r+1 \text{ is odd;} \\ 2\sum_{i=0}^{r-1}(d-1)^i & \text{if } g =: 2r \text{ is even.} \end{cases}$$

It is not difficult to prove that a graph of minimum degree δ and girth g has at least $n_0(\delta, g)$ vertices (Exercise 7). Interestingly, one can obtain the same bound for its average degree:

Theorem 1.3.4. (Alon, Hoory & Linial 2002)
Let G be a graph. If $d(G) \geqslant d \geqslant 2$ and $g(G) \geqslant g \in \mathbb{N}$ then $|G| \geqslant n_0(d, g)$.

One aspect of Theorem 1.3.4 is that it guarantees the existence of a short cycle compared with $|G|$. Using just the easy minimum degree version of Exercise 7, we get the following rather general bound:

[2.3.1] **Corollary 1.3.5.** *If $\delta(G) \geqslant 3$ then $g(G) < 2 \log |G|$.*

Proof. If $g := g(G)$ is even then

$$n_0(3, g) = 2\,\frac{2^{g/2} - 1}{2 - 1} = 2^{g/2} + (2^{g/2} - 2) > 2^{g/2},$$

while if g is odd then

$$n_0(3, g) = 1 + 3\,\frac{2^{(g-1)/2} - 1}{2 - 1} = \frac{3}{\sqrt{2}}\,2^{g/2} - 2 > 2^{g/2}.$$

As $|G| \geqslant n_0(3, g)$, the result follows. □

walk A *walk* (of *length* k) in a graph G is a non-empty alternating sequence $v_0 e_0 v_1 e_1 \ldots e_{k-1} v_k$ 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?).

1.4 Connectivity

connected A graph G is called *connected* if it is non-empty and 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). Instead of 'not connected' we usually say 'disconnected'.

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

[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. □

Let $G = (V, E)$ be a graph. A maximal connected subgraph of G is a *component* of G. Clearly, the components are induced subgraphs, and their vertex sets partition V. Since connected graphs are non-empty, the empty graph has no components. *component*

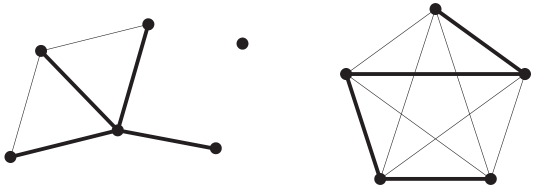


Fig. 1.4.1. A graph with three components, and a minimal spanning connected subgraph in each component

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. Note that this implies $A \cap B \subseteq X$. We say that X *separates* two vertices a, b if it separates the sets $\{a\}, \{b\}$ but $a, b \notin X$, and that X *separates* G if X separates some two vertices in G. A separating set of vertices is a *separator*. Separating sets of edges have no generic name, but some such sets do; see Section 1.9 for the definition of *cuts* and *bonds*. 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. *separate* *separator* *cutvertex* *bridge*

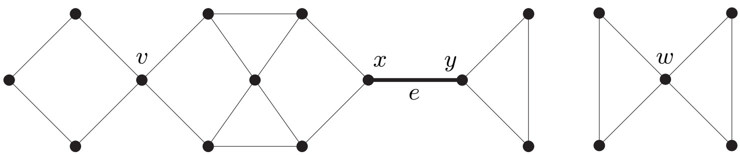


Fig. 1.4.2. A graph with cutvertices v, x, y, w and bridge $e = xy$

The unordered pair $\{A, B\}$ is a *separation* of G if $A \cup B = V$ and G has no edge between $A \setminus B$ and $B \setminus A$. Clearly, the latter is equivalent to saying that $A \cap B$ separates A from B. If both $A \setminus B$ and $B \setminus A$ are non-empty, the separation is *proper*. The number $|A \cap B|$ is the *order* of the separation $\{A, B\}$; the sets A, B are its *sides*. *separation*

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 *k-connected* 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 $n \geqslant 1$.

ℓ-edge-connected

edge-connectivity $\lambda(G)$

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 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.

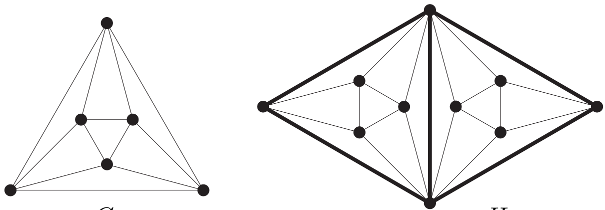


Fig. 1.4.3. The *octahedron* G (left) with $\kappa(G) = \lambda(G) = 4$, and a graph H with $\kappa(H) = 2$ but $\lambda(H) = 4$

[3.2.1] **Proposition 1.4.2.** *If G is non-trivial then $\kappa(G) \leqslant \lambda(G) \leqslant \delta(G)$.*

Proof. The second inequality follows from the fact that all the edges incident with a fixed vertex separate G. To prove the first, let F be a set of $\lambda(G)$ edges such that $G - F$ is disconnected. Such a set exists by definition of λ; note that F is a minimal separating set of edges in G. We show that $\kappa(G) \leqslant |F|$.

Suppose first that G has a vertex v that is not incident with an edge in F. Let C be the component of $G - F$ containing v. Then the vertices of C that are incident with an edge in F separate v from $G - C$. Since no edge in F has both ends in C (by the minimality of F), there are at most $|F|$ such vertices, giving $\kappa(G) \leqslant |F|$ as desired.

Suppose now that every vertex is incident with an edge in F. Let v be any vertex, and let C be the component of $G - F$ containing v. Then the neighbours w of v with $vw \notin F$ lie in C and are incident with distinct edges in F (again by the minimality of F), giving $d_G(v) \leqslant |F|$. As $N_G(v)$ separates v from any other vertices in G, this yields $\kappa(G) \leqslant |F|$—unless there are no other vertices, i.e. unless $\{v\} \cup N(v) = V$. But v was an arbitrary vertex. So we may assume that G is complete, giving $\kappa(G) = \lambda(G) = |G| - 1$. □

By Proposition 1.4.2, 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 1.4.3. (Mader 1972)
Let $0 \neq k \in \mathbb{N}$. Every graph G with $d(G) \geqslant 4k$ has a $(k+1)$-connected subgraph H such that $\varepsilon(H) > \varepsilon(G) - k$.

(1.2.2)
(1.3.1)

Proof. Put $\gamma := \varepsilon(G)$ $(\geqslant 2k)$, and consider the subgraphs $G' \subseteq G$ such that γ

$$|G'| \geqslant 2k \quad \text{and} \quad \|G'\| > \gamma\left(|G'| - k\right). \qquad (*)$$

Such graphs G' exist since G is one; let H be one of smallest order. H

No graph G' as in $(*)$ can have order exactly $2k$, since this would imply that $\|G'\| > \gamma k \geqslant 2k^2 > \binom{|G'|}{2}$. The minimality of H therefore implies that $\delta(H) > \gamma$: otherwise we could delete a vertex of degree at most γ and obtain a graph $G' \subsetneq H$ still satisfying $(*)$. In particular, we have $|H| \geqslant \gamma$. Dividing the inequality of $\|H\| > \gamma\,|H| - \gamma k$ from $(*)$ by $|H|$ therefore yields $\varepsilon(H) > \gamma - k$, as desired.

It remains to show that H is $(k+1)$-connected. If not, then H has a proper separation $\{U_1, U_2\}$ of order at most k; put $H[U_i] =: H_i$. H_1, H_2
Since any vertex $v \in U_1 \smallsetminus U_2$ has all its $d(v) \geqslant \delta(H) > \gamma$ neighbours from H in H_1, we have $|H_1| \geqslant \gamma \geqslant 2k$. Similarly, $|H_2| \geqslant 2k$. As by the minimality of H neither H_1 nor H_2 satisfies $(*)$, we further have

$$\|H_i\| \leqslant \gamma\left(|H_i| - k\right)$$

for $i = 1, 2$. But then

$$\begin{aligned} \|H\| &\leqslant \|H_1\| + \|H_2\| \\ &\leqslant \gamma\left(|H_1| + |H_2| - 2k\right) \\ &\leqslant \gamma\left(|H| - k\right) \qquad (\text{as } |H_1 \cap H_2| \leqslant k), \end{aligned}$$

which contradicts $(*)$ for H. □

1.5 Trees and forests

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*,[5] the others are its *inner vertices*. Every non-trivial tree has a leaf—consider, 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.

forest *tree* *leaf*

[5] ...except that the *root* of a tree (see below) is never called a leaf, even if it has degree 1.

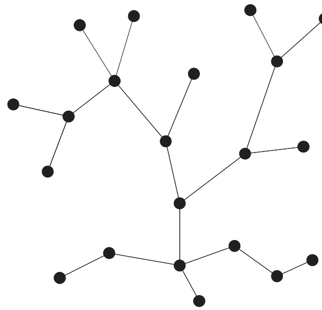


Fig. 1.5.1. A tree

[1.6.1]
[1.9.5]
[4.2.9]
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 common application of Theorem 1.5.1 is that every connected graph contains a spanning tree: take a minimal connected spanning subgraph and use (iii), or take a maximal acyclic subgraph and apply (iv). Figure 1.4.1 shows a spanning tree in each of the three components of the graph depicted. When T is a spanning tree of G, the edges in $E(G) \setminus E(T)$ are the *chords* of T in G.

chord

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 $\{v_1, \dots, v_{i-1}\}$.*

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

[1.9.5]
[2.4.4]
[4.2.9]
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 $G = G'$. □

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.* [9.2.1] [9.2.3]

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 T with a fixed root r is a *rooted tree*. Writing $x \leqslant y$ for $x \in rTy$ then defines a partial ordering on $V(T)$, the *tree-order* associated with T and r. We shall think of this ordering as expressing 'height': if $x < y$ we say that x lies *below* y in T, we call

root
tree-order
up/above
down/below

$$\lceil y \rceil := \{\, x \mid x \leqslant y \,\} \quad \text{and} \quad \lfloor x \rfloor := \{\, y \mid y \geqslant x \,\}$$

$\lceil t \rceil, \lfloor t \rfloor$

the *down-closure* of y and the *up-closure* of x, and so on. A set $X \subseteq V(T)$ that equals its up-closure, i.e. which satisfies $X = \lfloor X \rfloor := \bigcup_{x \in X} \lfloor x \rfloor$, is *closed upwards*, or an *up-set* in T. Similarly, there are *down-closed* sets, or *down-sets* etc..

down-closure
up-closure

Note that the root of T is the least element in its tree-order, the leaves are its maximal elements, the ends of any edge of T are comparable, and the down-closure of every vertex is a *chain*, a set of pairwise comparable elements. (Proofs?) The vertices at distance k from the root have *height* k and form the kth *level* of T.

chain
height, level

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 tree

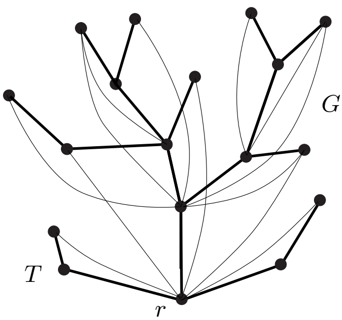


Fig. 1.5.2. A normal spanning tree with root r

A normal tree T in G can be a powerful tool for examining the structure of G, because G reflects the separation properties of T:

[8.2.3] [8.6.8] **Lemma 1.5.5.** *Let T be a normal tree in G.*

(i) *Any two vertices $x, y \in T$ are separated in G by the set $\lceil x \rceil \cap \lceil y \rceil$.*

(ii) *If $S \subseteq V(T) = V(G)$ and S is down-closed, then the components of $G - S$ are spanned by the sets $\lfloor x \rfloor$ with x minimal in $T - S$.*

Proof. (i) Let P be any x–y path in G; we show that P meets $\lceil x \rceil \cap \lceil y \rceil$. Let $t_1, \ldots, t_n$ be a minimal sequence of vertices in $P \cap T$ such that $t_1 = x$ and $t_n = y$ and t_i and t_{i+1} are comparable in the tree-order of T for all i. (Such a sequence exists: the set of *all* vertices in $P \cap T$, in their natural order as they occur on P, has this property because T is normal and every segment $t_i P t_{i+1}$ is either an edge of T or a T-path.) In our minimal sequence we cannot have $t_{i-1} < t_i > t_{i+1}$ for any i, since t_{i-1} and t_{i+1} would then be comparable, and deleting t_i would yield a smaller such sequence. Thus, our sequence has the form

$$x = t_1 > \ldots > t_k < \ldots < t_n = y$$

for some $k \in \{1, \ldots, n\}$. As $t_k \in \lceil x \rceil \cap \lceil y \rceil \cap V(P)$, our proof is complete.

(ii) Consider a component C of $G - S$, and let x be a minimal element of its vertex set. Then $V(C)$ has no other minimal element x': as x and x' would be incomparable, any x–x' path in C would by (i) contain a vertex below both, contradicting their minimality in $V(C)$. Hence as every vertex of C lies above some minimal element of $V(C)$, it lies above x. Conversely, every vertex $y \in \lfloor x \rfloor$ lies in C, for since S is down-closed, the ascending path xTy lies in $T - S$. Thus, $V(C) = \lfloor x \rfloor$.

Let us show that x is minimal not only in $V(C)$ but also in $T - S$. The vertices below x form a chain $\lceil t \rceil$ in T. As t is a neighbour of x, the maximality of C as a component of $G - S$ implies that $t \in S$, giving $\lceil t \rceil \subseteq S$ since S is down-closed. This completes the proof that every component of $G - S$ is spanned by a set $\lfloor x \rfloor$ with x minimal in $T - S$.

Conversely, if x is any minimal element of $T - S$, it is clearly also minimal in the component C of $G - S$ to which it belongs. Then $V(C) = \lfloor x \rfloor$ as before, i.e., $\lfloor x \rfloor$ spans this component. □

Normal spanning trees are also called *depth-first search trees*, because of the way they arise in computer searches on graphs (Exercise 26). This fact is often used to prove their existence, which can also be shown by a very short and clever induction (Exercise 25). The following constructive proof, however, illuminates better how normal trees capture the structure of their host graphs.

[6.5.3] [8.2.4] **Proposition 1.5.6.** *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 $V(T) = V(G)$.

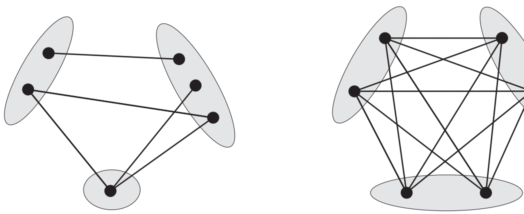
Suppose not, and let C be a component of $G - T$. As T is normal, $N(C)$ is a chain in T. Let x be its greatest element, and let $y \in C$ be adjacent to x. Let T' be the tree obtained from T by joining y to x; the tree-order of T' then extends that of T. We shall derive a contradiction by showing that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

1.6 Bipartite graphs

r-partite *bipartite*

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*.



$K_{2,2,2} = K_2^3$

Fig. 1.6.1. Two 3-partite graphs

complete r-partite $K_{n_1,\dots,n_r}$ K_s^r

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 complete r-partite graph $\overline{K^{n_1}} * \dots * \overline{K^{n_r}}$ is denoted by $K_{n_1,\dots,n_r}$; if $n_1 = \dots = n_r =: s$, 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.[6] (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

[6] 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.

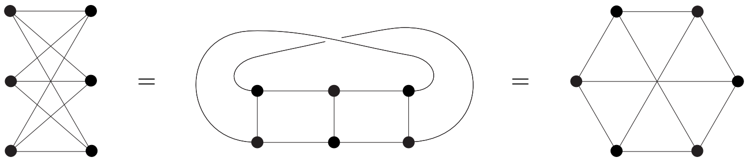


Fig. 1.6.2. Three drawings of the bipartite graph $K_{3,3} = K_3^2$

star *centre* called *stars*; the vertex in the singleton partition class of this $K_{1,n}$ is the star's *centre*.

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

[1.9.4] [5.3.1] [6.4.2] **Proposition 1.6.1.** *A graph is bipartite if and only if it contains no odd cycle.*

(1.5.1) *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.

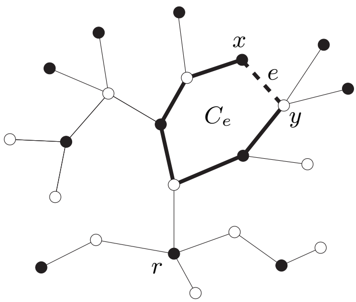


Fig. 1.6.3. The cycle C_e in $T + e$

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. $\square$

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 two more: the ‘minor’ relation, and the ‘topological minor’ relation. Let X be a fixed graph.

subdivision *TX of X*

branch vertices

A *subdivision* of X is, informally, any graph obtained from X by ‘subdividing’ some or all of its edges by drawing new vertices on those edges. In other words, we replace some edges of X with new paths between their ends, so that none of these paths has an inner vertex in $V(X)$ or on another new path. When G is a subdivision of X, we also say that G *is a* TX.[7] The original vertices of X are the *branch vertices of the* TX; its new vertices are called *subdividing vertices*. Note that subdividing vertices have degree 2, while branch vertices retain their degree from X.

topological minor

If a graph Y contains a TX as a subgraph, then X is a *topological minor* of Y (Fig. 1.7.1).

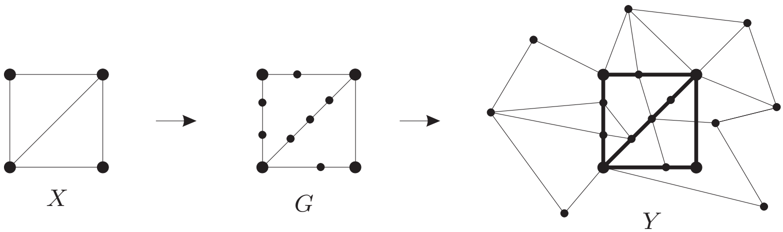


Fig. 1.7.1. The graph G is a TX, a *subdivision* of X.
As $G \subseteq Y$, this makes X a *topological minor* of Y.

IX

branch sets

contraction

Similarly, replacing the vertices x of X with disjoint connected graphs G_x, and the edges xy of X with non-empty sets of G_x–G_y edges, yields a graph that we shall call an IX.[8] More formally, a graph G *is an* IX if its vertex set admits a partition $\{ V_x \mid x \in V(X) \}$ into connected subsets V_x such that distinct vertices $x, y \in X$ are adjacent in X if and only if G contains a V_x–V_y edge. The sets V_x are the *branch sets of the* IX. Conversely, we say that X arises from G by *contracting* the subgraphs G_x and call it a *contraction minor* of Y.

minor, $\preccurlyeq$ *model*

If a graph Y contains an IX as a subgraph, then X is a *minor* of Y, the IX is a *model of* X in Y, and we write $X \preccurlyeq Y$ (Fig. 1.7.2).

---

[7] The ‘T’ stands for ‘topological’. Although, formally, TX denotes a whole class of graphs, the class of all subdivisions of X, it is customary to use the expression as indicated to refer to an arbitrary member of that class.

[8] The ‘I’ stands for ‘inflated’. As before, while IX is formally a class of graphs, those admitting a vertex partition $\{ V_x \mid x \in V(X) \}$ as described below, we use the expression as indicated to refer to an arbitrary member of that class.

Thus, X is a minor of Y if and only if there is a map φ from a subset of $V(Y)$ onto $V(X)$ such that for every vertex $x \in X$ its inverse image $\varphi^{-1}(x)$ is connected in Y and for every edge $xx' \in X$ there is an edge in Y between the branch sets $\varphi^{-1}(x)$ and $\varphi^{-1}(x')$ of its ends. If the domain of φ is all of $V(Y)$, and $xx' \in X$ whenever $x \neq x'$ and Y has an edge between $\varphi^{-1}(x)$ and $\varphi^{-1}(x')$ (so that Y is an IX), we call φ a

contraction *contraction* of Y onto X.

Since branch sets can be singletons, every subgraph of a graph is also its minor. In infinite graphs, branch sets are allowed to be infinite. For example, the graph shown in Figure 8.1.1 is an IX with X an infinite star.

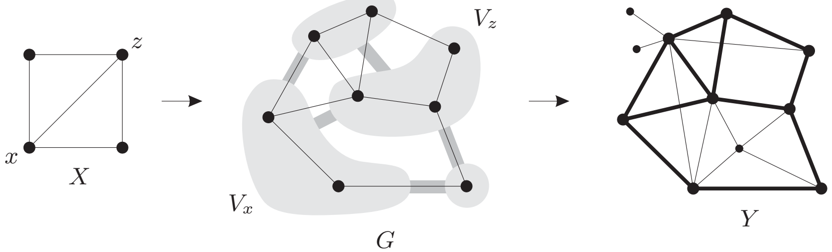


Fig. 1.7.2. The graph G is a model of X in Y, which makes X a *minor* of Y.

[12.6.1] **Proposition 1.7.1.** *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.* □

If G is an IX, then $P = \{ V_x \mid x \in X \}$ is a partition of $V(G)$, and we

G/P write $X =: G/P$ for this contraction minor of G. If $U = V_x$ is the only

G/U non-singleton branch set, we write $X =: G/U$, write v_U for the vertex

v_U $x \in X$ to which U contracts, and think of the rest of X as an induced subgraph of G. The 'smallest' non-trivial case of this is that U contains exactly two vertices forming an edge e, so that $U = e$. We then say that

contracting an edge $X = G/e$ arises from G by *contracting the edge* e; see Figure 1.7.3.

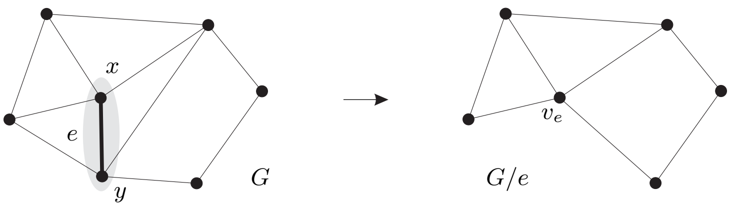


Fig. 1.7.3. Contracting the edge $e = xy$

Since the minor relation is transitive, every sequence of single vertex or edge deletions or contractions yields a minor. Conversely, every minor of a given finite graph can be obtained in this way:

Corollary 1.7.2. *Let X and Y be finite graphs. X is a minor of Y if and only if there are graphs $G_0, \dots, G_n$ such that $G_0 = Y$ and $G_n = X$ and each G_{i+1} arises from G_i by deleting an edge, contracting an edge, or deleting a vertex.*

Proof. Induction on $|Y| + \|Y\|$. □

Finally, we have the following relationship between minors and topological minors:

Proposition 1.7.3. [4.4.2] [7.3.1] [12.7.3]

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

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

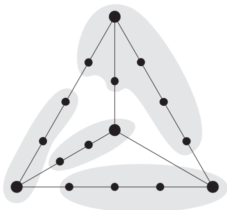


Fig. 1.7.4. A subdivision of K^4 viewed as an IK^4

Now that we have met all the standard relations between graphs, we can also define what it means to embed one graph in another. Basically, an *embedding* of G in H is an injective map $\varphi\colon V(G) \to V(H)$ that preserves the kind of structure we are interested in. Thus, φ embeds G in H 'as a subgraph' if it preserves the adjacency of vertices, and 'as an induced subgraph' if it preserves both adjacency and non-adjacency. If φ is defined on $E(G)$ as well as on $V(G)$ and maps the edges xy of G to independent paths in H between $\varphi(x)$ and $\varphi(y)$, it embeds G in H 'as a topological minor'. Similarly, an embedding φ of G in H 'as a minor' would be a map from $V(G)$ to disjoint connected vertex sets in H (rather than to single vertices) so that H has an edge between the sets $\varphi(x)$ and $\varphi(y)$ whenever xy is an edge of G. Further variants are possible; depending on the context, one may wish to define embeddings 'as a spanning subgraph', 'as an induced minor' and so on, in the obvious way. *embedding*

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

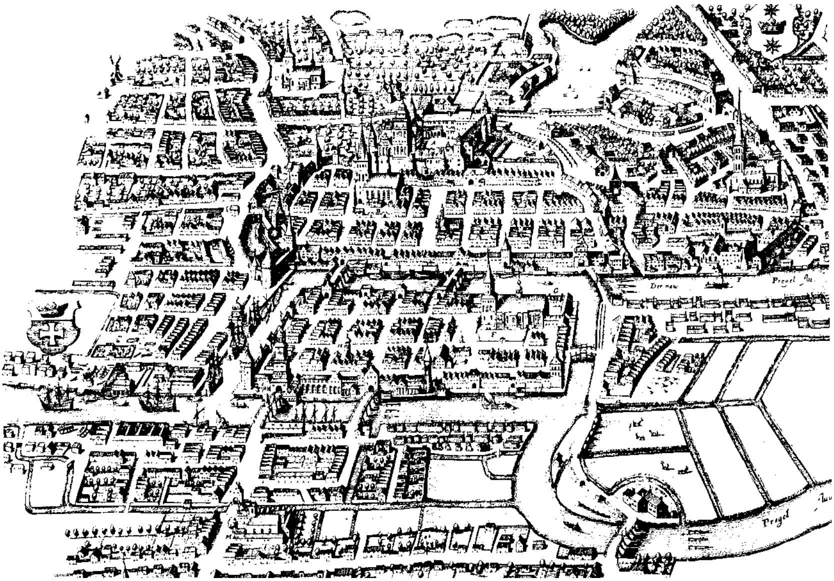


Fig. 1.8.1. The bridges of Königsberg (anno 1736)

Thus inspired,[9] 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

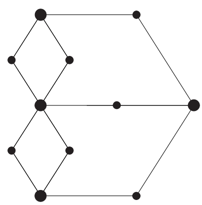


Fig. 1.8.2. A graph formalizing the bridge problem

[2.1.5] [10.3.1]

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

---

[9] 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.

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 $2k$.

Conversely, we show by induction on $\|G\|$ that every connected graph G with all degrees even has an Euler tour. The induction starts trivially with $\|G\| = 0$. Now let $\|G\| \geqslant 1$. Since all degrees are even, we can find in G a non-trivial closed walk that contains no edge more than once. (How exactly?) Let W be such a walk of maximal length, and write F for the set of its edges. If $F = E(G)$, then W is an Euler tour. Suppose, therefore, that $G' := G - F$ has an edge.

For every vertex $v \in G$, an even number of the edges of G at v lies in F, so the degrees of G' are again all even. Since G is connected, G' has an edge e incident with a vertex on W. By the induction hypothesis, the component C of G' containing e has an Euler tour. Concatenating this with W (suitably re-indexed), we obtain a closed walk in G that contradicts the maximal length of W. □

1.9 Some linear algebra [8.7]

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 $\dim \mathcal{V}(G) = n$.

$G = (V, E)$

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

$+$

In the same way as above, the functions $E \to \mathbb{F}_2$ form the *edge space* $\mathcal{E}(G)$ of G: its elements correspond to 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 $\dim \mathcal{E}(G) = m$. Given two elements F, F' of the edge space, viewed as functions $E \to \mathbb{F}_2$, we write

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

standard basis

$$\langle F, F' \rangle := \sum_{e \in E} F(e) F'(e) \in \mathbb{F}_2 \,.$$

$\langle F, F' \rangle$

This is zero if and only if F and F' have an even number of edges in common; in particular, we can have $\langle F, F \rangle = 0$ with $F \neq \emptyset$. Given a subspace $\mathcal{F}$ of $\mathcal{E}(G)$, we write

$\mathcal{F}^\perp$

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

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

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

cycle space $\mathcal{C}(G)$

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.[10] The dimension of $\mathcal{C}(G)$ is sometimes called the *cyclomatic number* of G.

The elements of $\mathcal{C}$ are easily recognized by the degrees of the subgraphs they form. Moreover, to generate the cycle space from cycles we only need disjoint unions rather than arbitrary symmetric differences:

[4.5.1] [8.7.3]

Proposition 1.9.1. *The following assertions are equivalent for edge sets $D \subseteq E$:*

(i) *$D \in \mathcal{C}(G)$;*

(ii) *D is a (possibly empty) disjoint union of edge sets of cycles in G;*

(iii) *All vertex degrees of the graph (V, D) are even.*

Proof. Since cycles have even degrees and taking symmetric differences preserves this, (i)→(iii) follows by induction on the number of cycles used to generate D. The implication (iii)→(ii) follows by induction on $|D|$: if $D \neq \emptyset$ then (V, D) contains a cycle C, whose edges we delete for the induction step. The implication (ii)→(i) is immediate from the definition of $\mathcal{C}(G)$. □

cut

A set F of edges is a *cut* in G if there exists a partition[11] $\{V_1, V_2\}$ of V such that $F = E(V_1, V_2)$. The edges in F are said to *cross* this partition. The sets V_1, V_2 are the *sides* of the cut. Recall that for $V_1 = \{v\}$ this cut is denoted by $E(v)$. A minimal non-empty cut in G is a *bond*.

bond

[4.6.3]

Proposition 1.9.2. *Together with $\emptyset$, the cuts in G form a subspace $\mathcal{B} = \mathcal{B}(G)$ of $\mathcal{E}(G)$. This space is generated by cuts of the form $E(v)$.*

Proof. Let $\mathcal{B}$ denote the subspace of $\mathcal{E}(G)$ generated by the cuts of the form $E(v)$. Every cut of G, with vertex partition $\{V_1, V_2\}$ say, equals $\sum_{v \in V_1} E(v)$ and hence lies in $\mathcal{B}$. Conversely, every set $\sum_{u \in U} E(u) \in \mathcal{B}$ is either empty, e.g. if $U \in \{\emptyset, V\}$, or it is the cut $E(U, V \setminus U)$. □

---

[10] For simplicity, we shall not always distinguish between the edge sets $F \in \mathcal{E}(G)$ and the subgraphs (V, F) they induce in G. When we wish to be more precise, such as in Chapter 8.6, we shall use the word '*circuit*' for the edge set of a cycle.

[11] Recall that partition classes in this book are non-empty. The empty set of edges, therefore, is a cut only if the graph is disconnected.

The space $\mathcal{B}$ from Proposition 1.9.2 is the *cut space*, or *bond space*, of G. It is not difficult to find among the cuts $E(v)$ an explicit basis for $\mathcal{B}$, and thus to determine its dimension (Exercise 40). Note that the bonds are for $\mathcal{B}$ what cycles are for $\mathcal{C}$: the minimal non-empty elements. *cut space* $\mathcal{B}(G)$

The 'non-empty' condition in the definition of a bond bites only if G is disconnected. If G is connected, its bonds are just its minimal cuts, and these are easy to recognize: a cut in a connected graph is minimal if and only if both sides of the corresponding vertex partition induce connected subgraphs (Exercise 36). If G is disconnected, its bonds are the minimal cuts of its components.

In analogy to Proposition 1.9.1, bonds and disjoint unions suffice to generate the cut space:

Lemma 1.9.3. *Every cut is a disjoint union of bonds.* [4.6.2] [6.5.2]

Proof. We apply induction on the size of the cut F considered. For $F = \emptyset$ the assertion is trivial (with the empty union). If $F \neq \emptyset$ is not itself a bond, it properly contains some other non-empty cut F'. By Proposition 1.9.2, also $F \setminus F' = F + F'$ is a smaller non-empty cut. By the induction hypothesis, both F' and $F \setminus F'$ are disjoint unions of bonds, and hence so is F. □

Exercise 39 indicates how to construct the bonds for Lemma 1.9.3 explicitly. In Chapter 3.1 we shall prove some more details about the possible positions of the cycles and bonds of a graph within its overall structure (Lemmas 3.1.2 and 3.1.3).

Theorem 1.9.4. *The cycle space $\mathcal{C}$ and the cut space $\mathcal{B}$ of any graph satisfy* [4.6]

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

Proof. Consider a graph $G = (V, E)$. Clearly, any cycle in G has an even number of edges in each cut. This implies $\mathcal{C} \subseteq \mathcal{B}^{\perp}$ and $\mathcal{B} \subseteq \mathcal{C}^{\perp}$. (1.6.1) (1.10)

To prove $\mathcal{B}^{\perp} \subseteq \mathcal{C}$, recall from Proposition 1.9.1 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 $E(v) \in \mathcal{B}$ implies $F \notin \mathcal{B}^{\perp}$. This completes the proof of $\mathcal{C} = \mathcal{B}^{\perp}$.

To prove $\mathcal{C}^{\perp} \subseteq \mathcal{B}$, let $F \in \mathcal{C}^{\perp}$ be given. Consider the multigraph[12] H obtained from G by contracting the edges in $E \setminus F$. Any cycle in H has all its edges in F. Since we can extend it to a cycle in G by edges from $E \setminus F$, the number of these edges must be even. Hence H is bipartite, by Proposition 1.6.1. Its bipartition induces a bipartition (V_1, V_2) of V such that $E(V_1, V_2) = F$, showing $F \in \mathcal{B}$ as desired. □

[12] See Section 1.10: such contractions might create loops in F, but bipartite multigraphs have no loops. The proof of Proposition 1.6.1 works for multigraphs too.

Consider a connected graph $G = (V, E)$ with a spanning tree $T \subseteq G$. For every chord $e \in E \setminus E(T)$ there is a unique cycle C_e in $T + e$, the *fundamental cycle* of e with respect to T. Similarly, for every edge $f \in T$ the forest $T - f$ has exactly two components (Theorem 1.5.1 (iii)). The set $D_f \subseteq E$ of edges of G between these components is a bond in G, the *fundamental cut* of f with respect to T.

fundamental cycle/cut

(1.5.1)

Notice that $f \in C_e$ if and only if $e \in D_f$, for all edges $e \notin T$ and $f \in T$. This is an indication of some deeper duality, which the following theorem explores further.

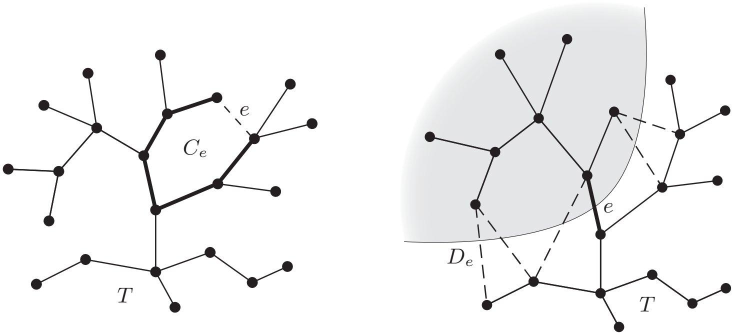


Fig. 1.9.2. The fundamental cycle C_e, and the fundamental cut D_f

[4.5.1] **Theorem 1.9.5.** *Let G be a connected graph with n vertices and m edges, and let $T \subseteq G$ a spanning tree.*

(i) *The fundamental cuts and cycles of G with respect to T form bases of $\mathcal{B}(G)$ and $\mathcal{C}(G)$, respectively.*

(ii) *Hence,* $\dim \mathcal{B}(G) = n - 1$ *and* $\dim \mathcal{C}(G) = m - n + 1$.

(1.5.3) *Proof.* (i) Note that an edge $f \in T$ lies in D_f but in no other fundamental cut, while an edge $e \notin T$ lies in C_e but in no other fundamental cycle. Hence the fundamental cuts and cycles form linearly independent sets in $\mathcal{B} = \mathcal{B}(G)$ and $\mathcal{C} = \mathcal{C}(G)$, respectively.

Let us show that the fundamental cycles generate every cycle C. By our initial observation, $D := C + \sum_{e \in C \setminus T} C_e$ is an element of $\mathcal{C}$ that contains no edge outside T. But by Proposition 1.9.1, the only element of $\mathcal{C}$ contained in T is $\emptyset$. So $D = \emptyset$, giving $C = \sum_{e \in C \setminus T} C_e$.

Similarly, every cut D is a sum of fundamental cuts. Indeed, the element $D + \sum_{f \in D \cap T} D_f$ of $\mathcal{B}$ contains no edge of T. As $\emptyset$ is the only element of $\mathcal{B}$ missing T, this implies $D = \sum_{f \in D \cap T} D_f$.

(ii) By (i), the fundamental cuts and cycles form bases of $\mathcal{B}$ and $\mathcal{C}$. As there are $n-1$ fundamental cuts (Corollary 1.5.3), there are $m-n+1$ fundamental cycles. □

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 *incidence matrix*

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

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. As is easy to check, B maps an edge set $F \subseteq E$ to the set of vertices incident with an odd number of edges in F, while B^t maps a set $U \subseteq V$ to set of edges with exactly one end in U. In particular:

Proposition 1.9.6.

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

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

More on this in the exercises and notes at the end of this chapter.

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

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

Viewed as a linear map $\mathcal{V} \to \mathcal{V}$, the adjacency matrix maps a given set $U \subseteq V$ to the set of vertices with an odd number of neighbours in U.

Let D denote the real diagonal matrix $(d_{ij})_{n\times n}$ with $d_{ii} = d(v_i)$ and $d_{ij} = 0$ otherwise. Our last proposition establishes a connection between A and B (now viewed as real matrices), which can be verified simply from the definition of matrix multiplication:

Proposition 1.9.7. $BB^t = A + D$. □

It is also instructive to check that $A + D$, with entries taken mod 2, defines the same map $\mathcal{V} \to \mathcal{V}$ as the composition of the maps of B and B^t (Exercise 48).

1.10 Other notions of graphs

For completeness, we now mention a few other notions of graphs which feature less frequently or not at all in this book.

A *hypergraph* is a pair (V, E) of disjoint sets, where the elements of E are non-empty subsets (of any cardinality) of V. Thus, graphs are special hypergraphs. *hypergraph*

A *directed graph* (or *digraph*) is a pair (V, E) of disjoint sets (of *vertices* and *edges*) together with two maps $\text{init}\colon E \to V$ and $\text{ter}\colon E \to V$ *directed graph*

init(e)
ter(e)

assigning to every edge e an *initial vertex* init(e) and a *terminal vertex* ter(e). The edge e is said to be *directed from* init(e) *to* ter(e). Note that a directed graph may have several edges between the same two vertices x, y. Such edges are called *multiple edges*; if they have the same direction (say from x to y), they are *parallel*. If $\text{init}(e) = \text{ter}(e)$, the edge e is called a *loop*.

loop

orientation

oriented graph

A directed graph D is an *orientation* of an (undirected) graph G if $V(D) = V(G)$ and $E(D) = E(G)$, and if $\{\text{init}(e), \text{ter}(e)\} = \{x, y\}$ for every edge $e = xy$. Intuitively, such an *oriented graph* arises from an undirected graph simply by directing every edge from one of its ends to the other. Put differently, oriented graphs are directed graphs without loops or multiple edges.

multigraph

A *multigraph* is a pair (V, E) of disjoint sets (of *vertices* and *edges*) together with a map $E \to V \cup [V]^2$ assigning to every edge either one or two vertices, its *ends*. Thus, multigraphs too can have loops and multiple edges: we may think of a multigraph as a directed graph whose edge directions have been 'forgotten'. To express that x and y are the ends of an edge e we still write $e = xy$, though this no longer determines e uniquely.

A graph is thus essentially the same as a multigraph without loops or multiple edges. Somewhat surprisingly, proving a graph theorem more generally for multigraphs may, on occasion, simplify the proof. Moreover, there are areas in graph theory (such as plane duality; see Chapters 4.6 and 6.5) where multigraphs arise more naturally than graphs, and where any restriction to the latter would seem artificial and be technically complicated. We shall therefore consider multigraphs in these cases, but without much technical ado: terminology introduced earlier for graphs will be used correspondingly.

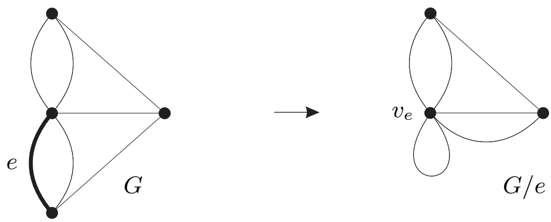


Fig. 1.10.1. Contracting the edge e in the multigraph corresponding to Fig. 1.8.1

A few differences, however, should be pointed out. A multigraph may have cycles of length 1 or 2: loops, and pairs of multiple edges (or *double edges*). A loop at a vertex makes it its own neighbour, and contributes 2 to its degree; in Figure 1.10.1, we thus have $d(v_e) = 6$. The ends of loops and parallel edges in a multigraph G are considered as separating that edge from the rest of G. The vertex v of a loop e, therefore, is a cutvertex unless $(\{v\},\{e\})$ is a component of G, and $(\{v\},\{e\})$ is a 'block' in the sense of Chapter 3.1. Thus, a multigraph with a loop is never 2-connected, and any 3-connected multigraph is in fact a graph.

The notion of edge contraction is simpler in multigraphs than in graphs. If we contract an edge $e = xy$ in a multigraph $G = (V, E)$ to a new vertex v_e, there is no longer a need to delete any edges other than e itself: edges parallel to e become loops at v_e, while edges xv and yv become parallel edges between v_e and v (Fig. 1.10.1). Thus, formally, $E(G/e) = E \smallsetminus \{e\}$, and only the incidence map $e' \mapsto \{\mathrm{init}(e'), \mathrm{ter}(e')\}$ of G has to be adjusted to the new vertex set in G/e. Contracting a loop thus has the same effect as deleting it.

The notion of a minor adapts accordingly. The contraction minor G/P defined by a partition P of $V(G)$ into connected sets has precisely those edges of G that join distinct partition classes. If there are several such edges between the same two classes, they become parallel edges of G/P. However, we do not normally give G/P any loops resulting from edges of G whose ends lie in the same partition class U. This would require us to say which of the edges of $G[U]$ are contracted (assuming they induce a connected spanning subgraph of $G[U]$), or at least how many are, which seems futile if we do not care about loops in G/P anyway.

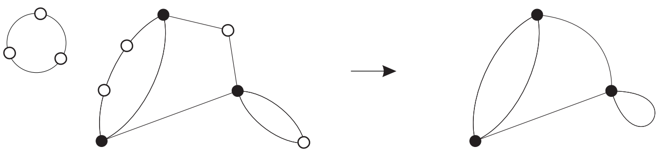


Fig. 1.10.2. Suppressing the white vertices

suppressing a vertex

If v is a vertex of degree 2 in a multigraph G, then by *suppressing* v we mean deleting v and adding an edge between its two neighbours.[13] (If its two incident edges are identical, i.e. form a loop at v, we add no edge and obtain just $G - v$. If they go to the same vertex $w \neq v$, the added edge will be a loop at w. See Figure 1.10.2.) Since the degrees of all vertices other than v remain unchanged when v is suppressed, suppressing several vertices of G always yields a well-defined multigraph that is independent of the order in which those vertices are suppressed.

Finally, it should be pointed out that authors who usually work with multigraphs tend to call them 'graphs'; in their terminology, our graphs would be called 'simple graphs'.

[13] This is just a clumsy combinatorial paraphrase of the topological notion of amalgamating the two edges at v into one edge, of which v becomes an inner point.

Exercises

1.$^-$ What is the number of edges in a K^n?

2. Let $d \in \mathbb{N}$ and $V := \{0,1\}^d$; thus, V is the set of all 0–1 sequences of length d. The graph on V in which two such sequences form an edge if and only if they differ in exactly one position is called the *d-dimensional cube*. Determine the average degree, number of edges, diameter, girth and circumference of this graph.

   (Hint for the circumference: induction on d.)

3. Let G be a graph containing a cycle C, and assume that G contains a path of length at least k between two vertices of C. Show that G contains a cycle of length at least $\sqrt{k}$.

4.$^-$ Is the bound in Proposition 1.3.2 best possible?

5. Let v_0 be a vertex in a graph G, and $D_0 := \{v_0\}$. For $n = 1, 2, \ldots$ inductively define $D_n := N_G(D_0 \cup \ldots \cup D_{n-1})$. Show that $D_n = \{ v \mid d(v_0, v) = n \}$ and $D_{n+1} \subseteq N(D_n) \subseteq D_{n-1} \cup D_{n+1}$ for all $n \in \mathbb{N}$.

6. Show that $\mathrm{rad}(G) \leqslant \mathrm{diam}(G) \leqslant 2\,\mathrm{rad}(G)$ for every graph G.

7. Prove the weakening of Theorem 1.3.4 obtained by replacing average with minimum degree. Deduce that $|G| \geqslant n_0(d/2, g)$ for every graph G as given in the theorem.

8. Show that graphs of girth at least 5 and order n have a minimum degree of $o(n)$. In other words, show that there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that $f(n)/n \to 0$ as $n \to \infty$ and $\delta(G) \leqslant f(n)$ for all such graphs G.

9.$^+$ Show that every connected graph G contains a path or cycle of length at least $\min\{2\delta(G), |G|\}$.

10. Show that a connected graph of diameter k and minimum degree d has at least about $kd/3$ vertices but need not have substantially more.

11.$^-$ Show that the components of a graph partition its vertex set. (In other words, show that every vertex belongs to exactly one component.)

12.$^-$ Show that every 2-connected graph contains a cycle.

13. Determine $\kappa(G)$ and $\lambda(G)$ for $G = P^m, C^n, K^n, K_{m,n}$ and the d-dimensional cube (Exercise 2); $d, m, n \geqslant 3$.

14.$^-$ Is there a function $f\colon \mathbb{N} \to \mathbb{N}$ such that, for all $k \in \mathbb{N}$, every graph of minimum degree at least $f(k)$ is k-connected?

15.$^+$ Let α, β be two graph invariants with positive integer values. Formalize the two statements below, and show that each implies the other:

   (i) β is bounded above by a function of α;

   (ii) α can be forced up by making β large enough.

   Show that the statement

   (iii) α is bounded below by a function of β

   is not equivalent to (i) and (ii). Which small change will make it so?