List of Figures

- GFDL: Gnu Free Documentation License. http://www.gnu.org/licenses/fdl.html
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. http://creativecommons.org/licenses/by-sa/3.0/
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. http://creativecommons.org/licenses/by-sa/2.5/
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. http://creativecommons.org/licenses/by-sa/1.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/deed.en
- cc-by-2.5: Creative Commons Attribution 2.5 License. http://creativecommons.org/licenses/by/2.5/deed.en
- cc-by-3.0: Creative Commons Attribution 3.0 License. http://creativecommons.org/licenses/by/3.0/deed.en
- GPL: GNU General Public License. http://www.gnu.org/licenses/gpl-2.0.txt
- LGPL: GNU Lesser General Public License. http://www.gnu.org/licenses/lgpl.html
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. http://artlibre.org/licence/lal/de
- CFR: Copyright free use.

FËUE WHEP

5552

- EPL: Eclipse Public License. http://www.eclipse.org/org/documents/epl-v10.php

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses[101]. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrower.

101 Chapter 36 on page 179

FĚUE WHEP

5553

1		GFDL
2	User:[102]	PD
3	Unknown, *The Hornet* is no longer in publication and it is very likely for a 20-year-old artist in 1871 to have died before 1939	PD
4		PD
5		GFDL
6		GFDL
7		PD
8	Uploader, CDC	GFDL
9	LadyofHats[103]	PD

102 http://en.wikibooks.org/wiki/User%3A
103 http://en.wikibooks.org/wiki/User%3ALadyofHats

FĚUE WHEP

FÊUE WHEP

36 Licenses

36.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10. 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

36.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title

Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

36.3 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

5558

Part V

WHEN IS ORDER DISORDER?

This section relates to the recent developments in chaos theory which pose significant questions for the historic scientific worldview, and it relates to the Christian doctrines of eschatology and the sovereignty of God. How does the indeterminacy of quantum physics relate to chaos? What are the implications of chaos for the idea that history is moving toward a culmination point?

What is the relationship between predictability and chaos? A great value of science is its ability to predict, to relate cause and effect. For example, eclipse dates can be calculated thousands of years into the past or future. There are systems that obey the laws of physics, yet generate random behavior. The roll of dice, the flow of a mountain stream, and the weather are all such phenomena; all have unpredictable aspects. Scientists now realize that simple deterministic systems can generate random behavior. Such behavior is called chaotic.

Is the physical order moving toward a culmination (eschatology)? How is the present reign of Christ (exaltation) over the created order to be understood? How does "chaos" relate to the free will/predestination debate?

Does the presence of chaos make a case for a dynamically involved Creator, such that the world is constantly emerging from chaos (Gen. 1:2)?

FĒUE WHEP

5559

Chapter Thirteen

CHAOS THEORY

In previous chapters we have dealt with the science of the very large (chapter 4—cosmology) and the very small (chapter 10—quantum theory). In this chapter we will deal with the science of everyday objects that cause us headaches. Here we will examine questions such as: Why does the ketchup not flow regularly from the bottle? Why can the dripping of the kitchen faucet be so irregular? Why cannot the weathercaster get the weather forecast correct? Why did the character Malcolm keep talking about chaos in the book and movie, *Jurassic Park?* The science of chaos deals with these frustrating and seemingly unrelated areas. In this chapter we will see why chaos theory has allowed scientists to discover order in certain apparently random processes. We will also examine the effects of chaos theory on our philosophical view of our world.

The Weather

As we discussed in chapter 10, for years astronomers have used Newton's laws to predict the future positions of planets or comets. Although the atmosphere has more particles than the solar system has planets, the same laws govern the behavior of the particles in the atmosphere. Meteorologists believed all that was needed was enough data to specify today's weather and a computer large enough to calculate the weather forecast for tomorrow. By the 1950s and 1960s, meteorologists were optimistic that they could achieve their goal of long-term weather forecasting. More and more weather stations—to collect temperature, pressure, and wind speed/direction—were being built; and large, powerful computers were becoming available. No one intended to do a calculation involving every particle in the atmosphere. Rather, they would

5560

model the atmosphere by including only those factors that are important in forecasting. This type of modeling is common in science. For example, when an astronomer calculates the path of Halley's Comet, she does not include every heavenly body in the calculation. She would not include other galaxies. She would only include the planets that cause the greatest effect on the comet's path.

In 1960 one person attempting to model the weather was the American meteorologist Edward N. Lorenz at the Massachusetts Institute of Technology. Lorenz had created a model of the weather involving twelve equations that related factors such as the temperature, pressure, and wind speed. Every minute his computer printed out a row of numbers that represented a day of weather. A review of the printout, line by line, gave the impression that his model was following earthly weather patterns. Pressure rose and fell; air currents swung north and south. One day he decided to repeat a set of calculations. In order to save time, he decided to start the calculations at the midpoint of the run. He entered the appropriate numbers from his computer printout and began the calculations.

Because the computer in his office was noisy, Lorenz left to get a cup of coffee. When he returned an hour later, he discovered, to his surprise, the results obtained were different from his first run. After just a few "months," all resemblance with the previous run was gone. How could this be? The results of this recalculation should have been the same since he was using the same program. After examining how his program worked, Lorenz realized that the computer used six-digit numbers (.506127) in its calculations. To save paper the printout was only to three digits (.506). The difference of one part in a thousand had resulted in vastly different behaviors for his system. Lorenz's finding was amazing since a scientist would usually consider himself lucky to reproduce two measurements with this level of precision. Lorenz had discovered a system that was very *sensitive to initial conditions*. Today we would say that Lorenz had discovered chaos.

To further analyze the behavior of systems sensitive to initial conditions, Lorenz decided to simplify his system. He developed a three-equation/three-variable system that did not model the weather but did model convection, a part of the atmosphere. *Convection* is the bulk movement of heat through a fluid. In his 1963 paper[1] Lorenz listed the output of his calculations: (0,10,0); (4,12,0); (9,20,0); (16,32,2); (30,66,7); (54,115,24); (93,192,74).

FEUE WHEP

5561

Lorenz obtained hundreds of these triplets. He wished to determine how the variables changed with time; one way to analyze this output is to graph the data which Lorenz did. Lorenz used each set of three numbers to represent a point in a three-dimensional space. The result of this plot would be a series of points. Connecting these points yields a continuous path which is a record of the system's behavior.

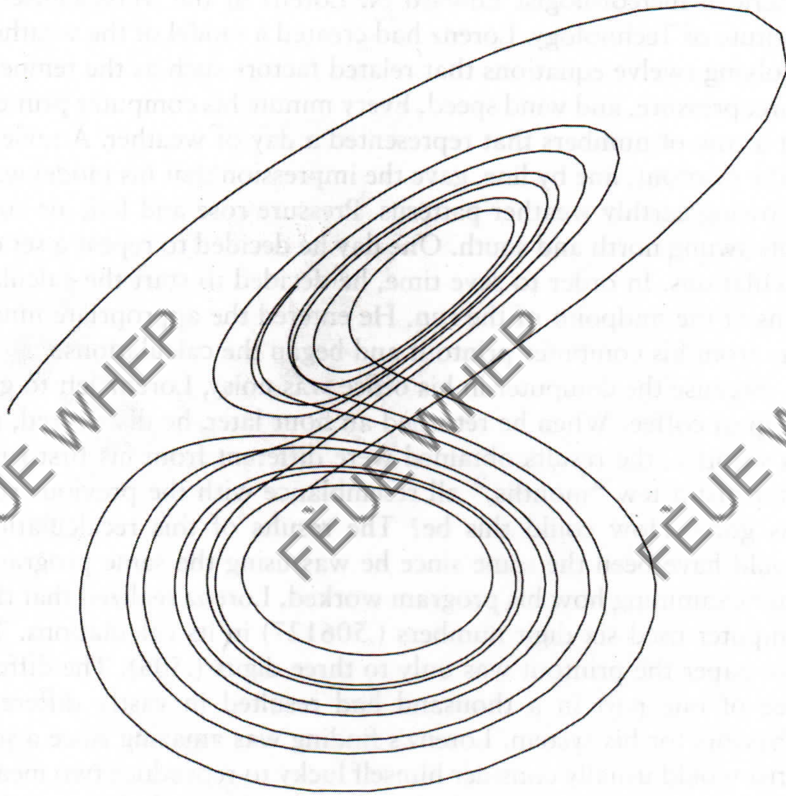


Fig. 13.1. Lorenz Attractor.

Lorenz discovered that the resulting pattern looks like an owl's face or the wings of a butterfly (see Fig. 13.1). The path weaves back and forth between the "wings," never repeating itself. The behavior signalled *disorder* since no path ever recurred. At the same time the behavior signalled *order* since all the paths were confined in the overall pattern. Since each set of initial conditions will result in a different path within the overall pattern, Lorenz concluded

FĚUE WHEP

5562

"that prediction of the sufficiently distant future is impossible by any method, unless the present conditions are known exactly. In view of the inevitable inaccuracy and incompleteness of weather observations, precise very-long-range forecasting would seem to be non-existent."[2] Thus, Lorenz is saying that because of the complexity of the atmosphere we can never have enough information to perform accurate weather forecasts.

Chaos

What is this chaos that Lorenz discovered in his weather model? How does it vary from everyday use of the words *chaos* and *random*? Looking up these words in the *Oxford English Dictionary* reveals that the word *chaos* comes from the ancient Greek concept of the original state of the universe as a formless void out of which the *cosmos* or order came. Thus, a chaotic state is one of utter confusion or disorder. The word *randomness* comes from an old French word meaning "to run fast" or "to gallop." Thus, something is random when it follows a haphazard course or is without aim or direction. Both of these words imply confusion or disorder.

Science also uses the terms *chaos* and *disorder,* along with the term *nonrandom*. In science these terms are distinguished by their degree of predictability. A *nonrandom* process is one that in theory and in practice allows predictability. When one thinks of the triumph of the scientific method, one is thinking of this predictability. Using the law of gravitation, one can predict eclipses thousands of years into the future or past. A *random* process is totally unpredictable. What has happened previously gives no clue as to what will happen next. Raindrops hitting a surface represent a random process because the arrival of one raindrop gives no clue to the arrival time of the next raindrop. A *chaotic* process falls in between these two extremes of total predictability and total unpredictability. Because equations can be written to describe the behavior of chaotic systems, they are in theory predictable. Yet, in practice, they are only temporarily predictable and eventually become unpredictable.

Attractors

How different is the behavior of Lorenz's system from other dynamic systems? *Dynamic* systems have constantly changing

5563

conditions in contrast to *static* systems. To obtain a picture of the behavior of the dynamic system, scientists graph the changing values of the systems variables. The resulting graph is called a *phase space,* which is a plot of the system over time. The phase space plot provides an idea of what the behavior of the system is like. With time, the graph will settle into a geometric shape called the *attractor*. The dynamic behavior is "attracted" to this geometric shape.

Fig. 13.2. Fixed-Point Attractor for a Playground Swing.

FEUE WHEP

5564

There are four kinds of attractors: fixed-point attractor, closed-curve attractor, torus attractor, and strange attractor. Figure 13.2 shows the behavior of a playground swing as it moves back and forth. Eventually, the swing comes to rest at a *fixed point,* its attractor. Start the swing again and it returns to this attractor.

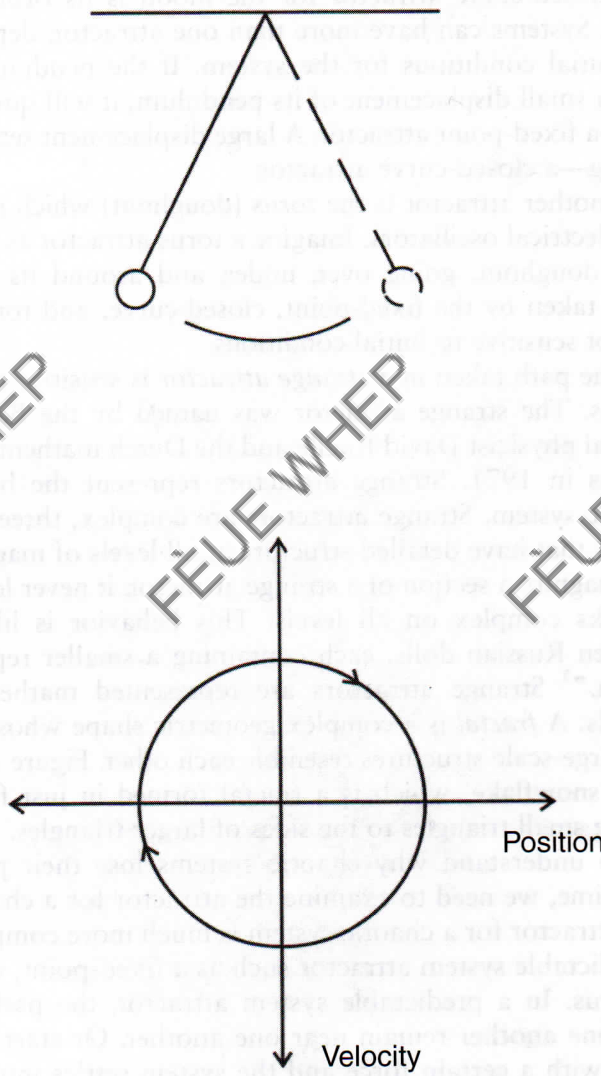


Fig. 13.3. Closed-Curve Attractor for a Pendulum Clock.

FĒUE WHEP

5565

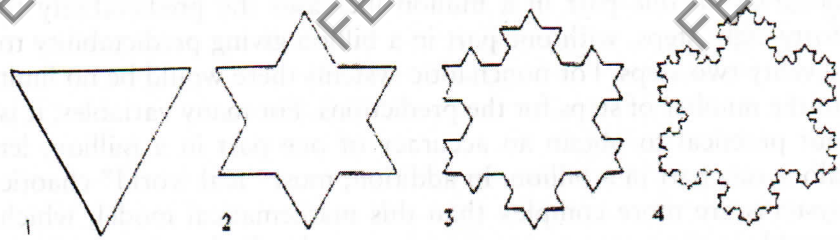
Drop a stone and it comes to rest at a fixed point on the earth, which is another example of a fixed-point attractor.

Not all attractors are fixed points. Some are cycles. A pendulum clock replaces its energy lost to friction by a spring or weight. Thus, the pendulum clock continuously repeats its swing. The attractor of the pendulum clock is a *closed curve* (see Fig. 13.3). The closed-curve attractor for the moon is its orbit around the earth. Systems can have more than one attractor, depending upon the initial conditions for the system. If the pendulum clock has only a small displacement of its pendulum, it will quickly come to rest—a fixed-point attractor. A large displacement sets the clock to ticking—a closed-curve attractor.

Another attractor is the *torus* (doughnut) which is seen in certain electrical oscillators. Imagine a torus attractor as walking on a large doughnut, going over, under, and around its surface. The paths taken by the fixed-point, closed-curve, and torus attractors are not sensitive to initial conditions.

The path taken in a *strange attractor* is sensitive to initial conditions. The strange attractor was named by the Belgian mathematical physicist David Ruelle and the Dutch mathematician Floris Takens in 1971. Strange attractors represent the behavior of a chaotic system. Strange attractors are complex, three-dimensional shapes that have detailed structure at all levels of magnification. If you magnify a section of a strange attractor, it never looks simpler; it looks complex on all levels. This behavior is like "a set of wooden Russian dolls, each containing a smaller replica of itself within."[3] Strange attractors are represented mathematically by fractals. A *fractal* is a complex geometric shape whose small-scale and large-scale structures resemble each other. Figure 13.4 shows a Koch snowflake, which is a fractal formed in just four steps by adding small triangles to the sides of larger triangles.

To understand why chaotic systems lose their predictability with time, we need to examine the attractor for a chaotic system. The attractor for a chaotic system is much more complicated than a predictable system attractor such as a fixed-point, closed curve, or torus. In a predictable system attractor, the paths that start near one another remain near one another. Or start a pendulum clock with a certain force and the system settles into the closed-curve attractor; change the starting force by a factor of one part in a thousand and the pendulum clock will settle into the "same"



A fractal is a geometric shape whose large-scale and small-scale structures resemble each other. An example of a fractal is the Koch snowflake which can be made by adding small triangles to the sides of larger triangles.

Fig. 13.4. Example of a Fractal.

closed-curve attractor. Thus, the system is not very sensitive to initial conditions, and predictability is maintained.

Different or "strange" behavior is observed for a chaotic system. With a strange attractor, the paths that start near one another quickly diverge. It is as if the attractor space is being stretched; a model of this behavior could be the stretching of a lump of bread dough during kneading. This divergence of the paths is exponential. An attractor is finite and thus the paths cannot diverge forever. Thus, the attractor folds onto itself; again, this can be modeled by the folding of bread dough after it has been stretched. The paths of the attractor are shuffled by the folding. The stretching and folding of the paths makes the system very dependent upon initial conditions. Now one can imagine why a one-part-in-a-thousand change in initial conditions caused the system to follow a different path as Lorenz observed. The stretching and folding continues repeatedly, creating a fractal. Also the stretching and folding of the paths replaces the initial information with new information. Predictability is short-lived. Long-term, all causality is lost.

Just how sensitive are chaotic systems to initial conditions? Before the concept of chaotic systems, it was assumed that all systems are predictable and that the accuracy of the prediction depended on the accuracy of the measurements of the system variables. Chaotic systems changed this view. As an example, for a very simple mathematical model of a chaotic process, measuring the values of the variables to one part in a thousand allows one only to predict the sequence of events for twenty-four steps. Increasing the

5567

accuracy to one part in a million increases the predictability to forty-eight steps, with one part in a billion giving predictability to seventy-two steps. For nonchaotic systems there would be no limit to the number of steps for the predictions. For many variables, it is not practical to obtain an accuracy of one part in a million, let alone one part in a billion. In addition, most "real world" chaotic systems are more complex than this mathematical model, which would require even greater accuracy to maintain these levels of predictability. In all chaotic cases, one quickly comes to the point where predictability breaks down.

Transition to Chaos

Many dynamic systems begin as ordered, predictable systems and then change to a chaotic system. We remember a scene from a movie where a character puts a lit cigarette on an ashtray. The camera focuses on the smoke. Initially, the smoke rises in a smooth stream (order). Then, suddenly, the top of the smooth stream becomes wildly erratic and swirls in all directions. The behavior of the smoke has gone from *laminar* (order) to *turbulent* (chaos) flow. Other examples of the transition to turbulence from our everyday life occur when a regularly dripping faucet changes to a randomly dripping faucet, or when ketchup smoothly flowing from a bottle onto our fries suddenly acquires an erratic flow and lands on us as well as the fries.

Turbulence has always been lurking in scientific systems. In many cases it could be ignored, so the systems were studied as ordered. In others, turbulence could not be ignored, so the systems were ignored. Engineers hate turbulence. Turbulent airflow removes the lift from an aircraft's wing; turbulent oil flow in a pipeline causes drag. Until the advent of chaos theory, few thought that turbulence would ever be understood. A scientific myth says that the quantum physicist Werner Heisenberg said that he had two questions for God: why relativity, and why turbulence? Heisenberg is quoted as saying, "I really think He may have an answer to the first question."[4]

An important question is how flow can change from smooth to turbulent. Or more generally, how do ordered systems become chaotic? Is there a way to predict when this transition will occur? One clue to understanding this transition came from the American biologist Robert May, who was studying annual variations in insect populations. One might expect that a high growth rate would lead

5568

to a larger population while a low growth rate would lead to a smaller population, with extinction occurring if the growth rate is too small. Using a mathematical model which predicted next year's population based on this year's population, May studied the effect of an increasing growth rate on the population value. At low values for the growth rate, the population would settle down to a single value year after year.

At first increasing the growth rate increased the population to another stable value. Then a surprise happened. As soon as the growth rate passed a value of three in his model, the possible population value branched (*bifurcated*) into two solutions. The model was still being deterministic (predicting a solution); it was now predicting two solutions. At this new growth rate, the population would be at one value one year followed by the other value the next year; then the population values would repeat. Increasing the growth rate a little more caused the population value choices to jump from two to four; the model was still deterministic. Continuing to increase the growth rate would lead to four bifurcating to eight, eight into sixteen, and so on with the model still being deterministic. At a value of 3.57, chaos began in his model. At this point, it became impossible to predict future population values; all one could say is that the population value would be one among all the values in the strange attractor. Field biologists found that May's model did reflect the behavior of actual animal populations.

Is May's work applicable to other systems that make a transition from predictable to chaotic? An answer came from the American physicist Mitchell Feigenbaum,[5] who studied May's work and proposed that the transition to chaos involves what is called period doubling (the bifurcation that May observed). This is called the *period-doubling route to chaos*. It was also realized that the period doubling comes faster and faster until the sudden appearance of chaos. Feigenbaum determined a numerical constant (4.669) that governs the doubling process (*Feigenbaum number*). He also discovered that these results (period doubling and Feigenbaum number) were applicable to a wide variety of chaotic systems. At last science had a way to predict the onset of turbulence.

Applications

As scientists acquired an understanding of chaotic process, they realized some processes that they thought were random were

5569

actually chaotic. In some cases, the chaotic model allowed scientists to explain puzzling observations. In other cases, scientists now had a tool for predicting the beginning of chaos and thereby at last being able to prevent the beginning of chaos. A few examples of the application of chaos theory to problems in astronomy and medicine will be given.

Astronomy

In astronomy, scientists have mostly used chaos theory to explain observations. Certain regions of the asteroid belt between Mars and Jupiter are almost free of asteroids. Scientists assumed that the gravitational field of Jupiter had resulted in these gaps; but until the advent of chaos theory, scientists had no mathematical model for these gaps (called *Kirkwood gaps* in honor of their discoverer). Calculations, using chaos theory, show that the interactions between the motions of the asteroids and the motion and gravitation field of Jupiter create chaotic regions in the asteroid belt. Most of the asteroids are expelled from these chaotic regions, resulting in the Kirkwood gaps. The expelled asteroids are sent on a path that takes them toward the inner planets. Thus, some of the expelled asteroids cross the orbit of Earth; such asteroids have the potential for colliding with the Earth and causing great damage. Chaos theory gave scientists a framework to explain and tie together these two phenomena.

Most natural satellites (moons) in the solar system have an orbit period equal to its spin period. For example, the moon takes twenty-seven days to orbit the earth and twenty-seven days to rotate on its axis. This results in the same side of the moon always facing the earth. Hyperion is a potato-shaped satellite of Saturn. Hyperion has an erratic spin period and a constantly changing rotational period. Calculations, using chaos theory, indicate that the behavior of Hyperion is chaotic because of its interaction with Saturn and Saturn's large moon, Titan.

Medicine

In medicine, chaos theory has not only given scientists an understanding of why certain phenomena occur but in some cases a regimen for preventing the onset of certain conditions. The body's defense mechanism against disease has been analyzed as a chaotic process. When the body is invaded by a bacterium or virus, the body, apparently, tries defense strategies at random. A

5570

feedback loop is used to tell (indicate) when a correct strategy has been selected. Scientists are trying to mimic this process in drug development.

Analysis of historical data for the two childhood diseases, measles and chicken pox, revealed that their epidemics behaved differently. Chicken pox varied periodically, while measles varied chaotically. This means that at a certain number of measles cases, it is "impossible" to predict in which direction the epidemic will proceed. The strange attractor for the measles outbreaks helps epidemiologists see patterns in what had previously been "random and noisy" yearly data.

Several medical conditions, including heart fibrillation and attacks of epilepsy and manic depression, involve a transition from an orderly process to a chaotic process. Fibrillation is also called ventricular fibrillation. The *ventricles* are the two large pumping chambers of the heart that discharge blood to the lungs or body. The normal ventricular contractions of the heart are periodic, controlled, and coordinated. *Fibrillation* involves ventricular contractions that are rapid, uncontrolled, and uncoordinated. Under these conditions the ventricles cannot pump blood and death can occur unless the condition is corrected. Normal heartbeat can sometimes be restored by a massive electric shock to the chest using a *defibrillator*. Chaos theory has helped physicians in two ways. The period-doubling-to-chaos trend allows those monitoring a patient to detect the beginning of the transition to chaos and to intervene before fibrillation starts. Understanding of the strange attractor for fibrillation has allowed for the better design of defibrillators.

The Scientific Method and Chaos

As we saw in chapter 1, one approach to the scientific method is to verify a theory by testing predictions. We compare the flight of a ball predicted by a mathematical equation to the actual flight of the ball. For more complicated systems such as a collection of gas molecules, a scientist would use statistical techniques to examine the properties of the system rather than the properties of the individual gas molecules. The system properties are statistical averages of the properties of the individual molecules. Simple mathematical equations relating the system's temperature, pressure, and volume can be found. (Reducing the volume increases the temperature and pressure of the system.) No attempt is made to explain the variables for an individual gas molecule.

5571

What about chaotic systems? Very short-term predictions are possible; in billiards, the agreement between the predicted and actual path of a cue ball is lost in a minute or less. Long-term predictions for chaotic systems are impossible. Long-term, the best one can hope for is agreement with the strange attractors. Even with this agreement, chaos is not explaining very much more than statistical techniques. Neither method is giving predictive information about the individual components.

Chaos challenges *reductionism*. Reductionism says that the whole can be understood by breaking it down and studying its parts; if one can determine the forces and components present, then one knows everything about the whole. This view has been very successful in physics and chemistry. Many scientists are now attempting to extend reductionism to biology; they believe if one can determine all the physical interactions and chemical reactions present in a living system, then one can totally explain that system. Chaos has shown that complex behavior arises from simple, nonlinear interactions of the system's components. This implies that the whole can be more complex than the sum of its parts.

Usually we emphasize the limitations of chaos, the loss of predictability. However, others have speculated about the positive effects of chaos in nature. It has been proposed that nature, by amplifying small fluctuations, creates novelty. Scientists wonder: Do prey use chaotic flight controls to evade predators? Does a chaotic process introduce genetic variability? Does creativity have an underlying chaotic process?

Summary

Chaos is not the same as randomness. Although chaotic systems are deterministic with mathematical equations that relate the behavior of their components, they lose long-term predictability. Chaotic systems have three characteristics: sensitivity to initial conditions, strange attractors, and period-doubling route to chaos. Chaos theory challenges the predictability that undergirds the scientific method; it also challenges reductionism. Although chaos may be seen as a limitation to our understanding of nature, chaos may be the mechanism by which novelty is introduced.

FEUE WHEP

5572

GRAPH THEORY WITH APPLICATIONS

J. A. Bondy and U. S. R. Murty

Department of Combinatorics and Optimization,
University of Waterloo,
Ontario, Canada

NORTH-HOLLAND
New York • Amsterdam • Oxford

FĒUE WHEP

5573

©J.A. Bondy and U.S.R. Murty 1976

First published in Great Britain 1976 by
The Macmillan Press Ltd.

First published in the U.S.A. 1976 by
Elsevier Science Publishing Co., Inc.
52 Vanderbilt Avenue, New York, N.Y. 10017

Fifth Printing, 1982.

Sole Distributor in the U.S.A:
Elsevier Science Publishing Co., Inc.

Library of Congress Cataloging in Publication Data

Bondy, John Adrian.
Graph theory with applications.

Bibliography: p.
Includes index.
1. Graph theory. I. Murty, U.S.R., joint author. II. Title.
QA166.B67 1979 511′.5 75-29826
ISBN 0-444-19451-7

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission.

Printed in the United States of America

5574

To our parents

FÈUE WHEP

5575

FEUE WHEP

Preface

This book is intended as an introduction to graph theory. Our aim has been to present what we consider to be the basic material, together with a wide variety of applications, both to other branches of mathematics and to real-world problems. Included are simple new proofs of theorems of Brooks, Chvátal, Tutte and Vizing. The applications have been carefully selected, and are treated in some depth. We have chosen to omit all so-called 'applications' that employ just the language of graphs and no theory. The applications appearing at the end of each chapter actually make use of theory developed earlier in the same chapter. We have also stressed the importance of efficient methods of solving problems. Several good algorithms are included and their efficiencies are analysed. We do not, however, go into the computer implementation of these algorithms.

The exercises at the end of each section are of varying difficulty. The harder ones are starred (*) and, for these, hints are provided in appendix I. In some exercises, new definitions are introduced. The reader is recommended to acquaint himself with these definitions. Other exercises, whose numbers are indicated by bold type, are used in subsequent sections; these should all be attempted.

Appendix II consists of a table in which basic properties of four graphs are listed. When new definitions are introduced, the reader may find it helpful to check his understanding by referring to this table. Appendix III includes a selection of interesting graphs with special properties. These may prove to be useful in testing new conjectures. In appendix IV, we collect together a number of unsolved problems, some known to be very difficult, and others more hopeful. Suggestions for further reading are given in appendix V.

Many people have contributed, either directly or indirectly, to this book. We are particularly indebted to C. Berge and D. J. A. Welsh for introducing us to graph theory, to G. A. Dirac, J. Edmonds, L. Lovász and W. T. Tutte, whose works have influenced our treatment of the subject, to V. Chungphaisan and C. St. J. A. Nash-Williams for their careful reading of the

5576

manuscript and valuable suggestions, and to the ubiquitous G. O. M. for his kindness and constant encouragement.

We also wish to thank S. B. Maurer, P. J. O'Halloran, C. Thomassen, B. Toft and our colleagues at the University of Waterloo for many helpful comments, and the National Research Council of Canada for its financial support. Finally, we would like to express our appreciation to Joan Selwood for her excellent typing and Diana Rajnovich for her beautiful artwork.

J. A. Bondy
U. S. R. Murty

FEUE WHEP

5577

Contents

5578

5579

FËUE WHEP

5580

1 Graphs and Subgraphs

1.1 GRAPHS AND SIMPLE GRAPHS

Many real-world situations can conveniently be described by means of a diagram consisting of a set of points together with lines joining certain pairs of these points. For example, the points could represent people, with lines joining pairs of friends; or the points might be communication centres, with lines representing communication links. Notice that in such diagrams one is mainly interested in whether or not two given points are joined by a line; the manner in which they are joined is immaterial. A mathematical abstraction of situations of this type gives rise to the concept of a graph.

A *graph* G is an ordered triple $(V(G), E(G), \psi_G)$ consisting of a nonempty set $V(G)$ of *vertices*, a set $E(G)$, disjoint from $V(G)$, of *edges*, and an *incidence function* ψ_G that associates with each edge of G an unordered pair of (not necessarily distinct) vertices of G. If e is an edge and u and v are vertices such that $\psi_G(e) = uv$, then e is said to *join* u and v; the vertices u and v are called the *ends* of e.

Two examples of graphs should serve to clarify the definition.

Example 1

$$G = (V(G), E(G), \psi_G)$$

where

$$V(G) = \{v_1, v_2, v_3, v_4, v_5\}$$
$$E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$$

and ψ_G is defined by

$$\psi_G(e_1) = v_1v_2,\ \psi_G(e_2) = v_2v_3,\ \psi_G(e_3) = v_3v_3,\ \psi_G(e_4) = v_3v_4$$
$$\psi_G(e_5) = v_2v_4,\ \psi_G(e_6) = v_4v_5,\ \psi_G(e_7) = v_2v_5,\ \psi_G(e_8) = v_2v_5$$

Example 2

$$H = (V(H), E(H), \psi_H)$$

where

$$V(H) = \{u, v, w, x, y\}$$
$$E(H) = \{a, b, c, d, e, f, g, h\}$$

and ψ_H is defined by

$$\psi_H(a) = uv,\quad \psi_H(b) = uu,\quad \psi_H(c) = vw,\quad \psi_H(d) = wx$$
$$\psi_H(e) = vx,\quad \psi_H(f) = wx,\quad \psi_H(g) = ux,\quad \psi_H(h) = xy$$

5581

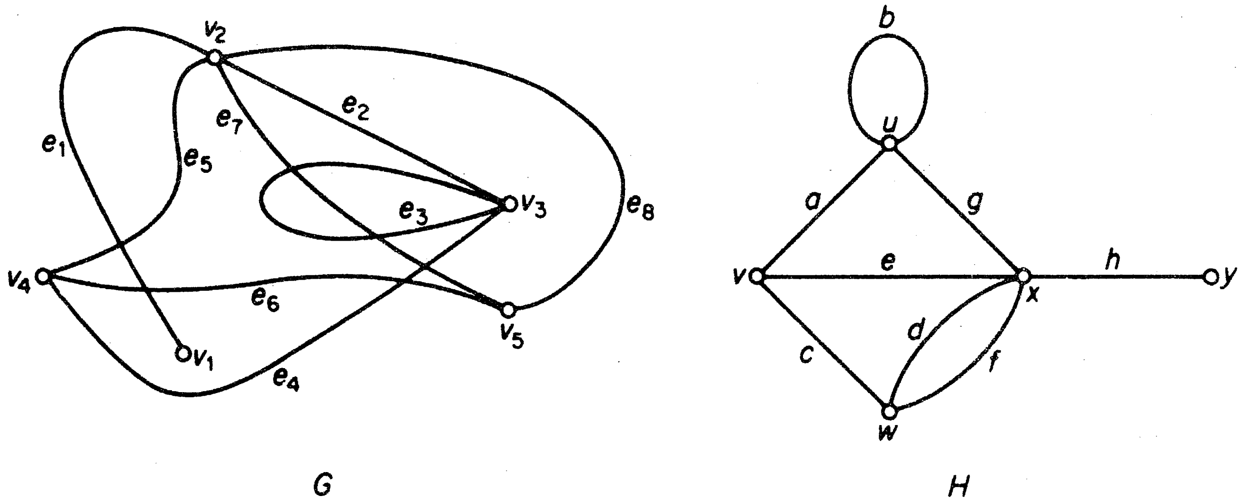


Figure 1.1. Diagrams of graphs G and H

Graphs are so named because they can be represented graphically, and it is this graphical representation which helps us understand many of their properties. Each vertex is indicated by a point, and each edge by a line joining the points which represent its ends.† Diagrams of G and H are shown in figure 1.1. (For clarity, vertices are depicted here as small circles.)

There is no unique way of drawing a graph; the relative positions of points representing vertices and lines representing edges have no significance. Another diagram of G, for example, is given in figure 1.2. A diagram of a graph merely depicts the incidence relation holding between its vertices and edges. We shall, however, often draw a diagram of a graph and refer to it as the graph itself; in the same spirit, we shall call its points 'vertices' and its lines 'edges'.

Note that two edges in a diagram of a graph may intersect at a point that

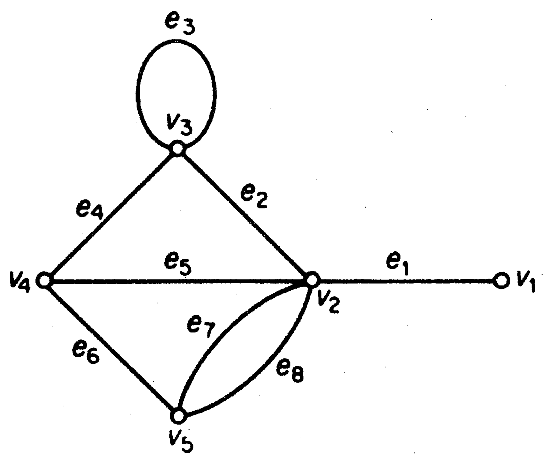


Figure 1.2. Another diagram of G

† In such a drawing it is understood that no line intersects itself or passes through a point representing a vertex which is not an end of the corresponding edge—this is clearly always possible.

5582

is not a vertex (for example e_1 and e_6 of graph G in figure 1.1). Those graphs that have a diagram whose edges intersect only at their ends are called *planar*, since such graphs can be represented in the plane in a simple manner. The graph of figure 1.3*a* is planar, even though this is not immediately clear from the particular representation shown (see exercise 1.1.2). The graph of figure 1.3*b*, on the other hand, is nonplanar. (This will be proved in chapter 9.)

Most of the definitions and concepts in graph theory are suggested by the graphical representation. The ends of an edge are said to be *incident* with the edge, and vice versa. Two vertices which are incident with a common edge are *adjacent*, as are two edges which are incident with a common vertex. An edge with identical ends is called a *loop*, and an edge with distinct ends a *link*. For example, the edge e_3 of G (figure 1.2) is a loop; all other edges of G are links.

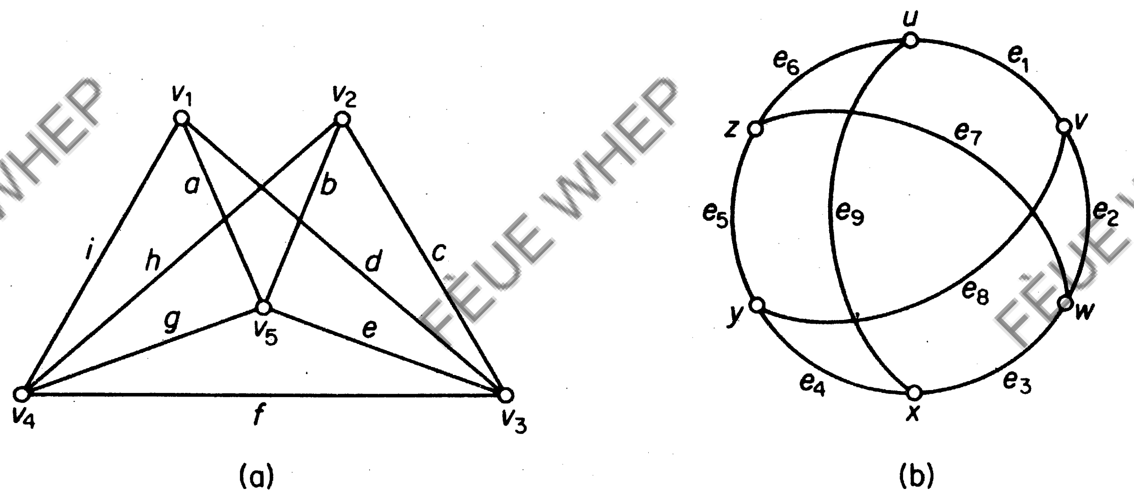


Figure 1.3. Planar and nonplanar graphs

A graph is *finite* if both its vertex set and edge set are finite. In this book we study only finite graphs, and so the term 'graph' always means 'finite graph'. We call a graph with just one vertex *trivial* and all other graphs *nontrivial*.

A graph is *simple* if it has no loops and no two of its links join the same pair of vertices. The graphs of figure 1.1 are not simple, whereas the graphs of figure 1.3 are. Much of graph theory is concerned with the study of simple graphs.

We use the symbols $\nu(G)$ and $\varepsilon(G)$ to denote the numbers of vertices and edges in graph G. Throughout the book the letter G denotes a graph. Moreover, when just one graph is under discussion, we usually denote this graph by G. We then omit the letter G from graph-theoretic symbols and write, for instance, V, E, ν and ε instead of $V(G)$, $E(G)$, $\nu(G)$ and $\varepsilon(G)$.

5583

Exercises

1.1.1 List five situations from everyday life in which graphs arise naturally.

1.1.2 Draw a different diagram of the graph of figure 1.3*a* to show that it is indeed planar.

1.1.3 Show that if G is simple, then $\varepsilon \le \binom{\nu}{2}$.

1.2 GRAPH ISOMORPHISM

Two graphs G and H are *identical* (written $G = H$) if $V(G) = V(H)$, $E(G) = E(H)$, and $\psi_G = \psi_H$. If two graphs are identical then they can clearly be represented by identical diagrams. However, it is also possible for graphs that are not identical to have essentially the same diagram. For example, the diagrams of G in figure 1.2 and H in figure 1.1 look exactly the same, with the exception that their vertices and edges have different labels. The graphs G and H are not identical, but isomorphic. In general, two graphs G and H are said to be *isomorphic* (written $G \cong H$) if there are bijections $\theta : V(G) \to V(H)$ and $\phi : E(G) \to E(H)$ such that $\psi_G(e) = uv$ if and only if $\psi_H(\phi(e)) = \theta(u)\theta(v)$; such a pair (θ, ϕ) of mappings is called an *isomorphism* between G and H.

To show that two graphs are isomorphic, one must indicate an isomorphism between them. The pair of mappings (θ, ϕ) defined by

$$\theta(v_1) = y, \quad \theta(v_2) = x, \quad \theta(v_3) = u, \quad \theta(v_4) = v, \quad \theta(v_5) = w$$

and

$$\begin{array}{llll} \phi(e_1) = h, & \phi(e_2) = g, & \phi(e_3) = b, & \phi(e_4) = a \\ \phi(e_5) = e, & \phi(e_6) = c, & \phi(e_7) = d, & \phi(e_8) = f \end{array}$$

is an isomorphism between the graphs G and H of examples 1 and 2; G and H clearly have the same structure, and differ only in the names of vertices and edges. Since it is in structural properties that we shall primarily be interested, we shall often omit labels when drawing graphs; an unlabelled graph can be thought of as a representative of an equivalence class of isomorphic graphs. We assign labels to vertices and edges in a graph mainly for the purpose of referring to them. For instance, when dealing with simple graphs, it is often convenient to refer to the edge with ends u and v as 'the edge uv'. (This convention results in no ambiguity since, in a simple graph, at most one edge joins any pair of vertices.)

We conclude this section by introducing some special classes of graphs. A simple graph in which each pair of distinct vertices is joined by an edge is called a *complete graph*. Up to isomorphism, there is just one complete graph on n vertices; it is denoted by K_n. A drawing of K_5 is shown in figure 1.4*a*. An *empty graph*, on the other hand, is one with no edges. A *bipartite*

5584

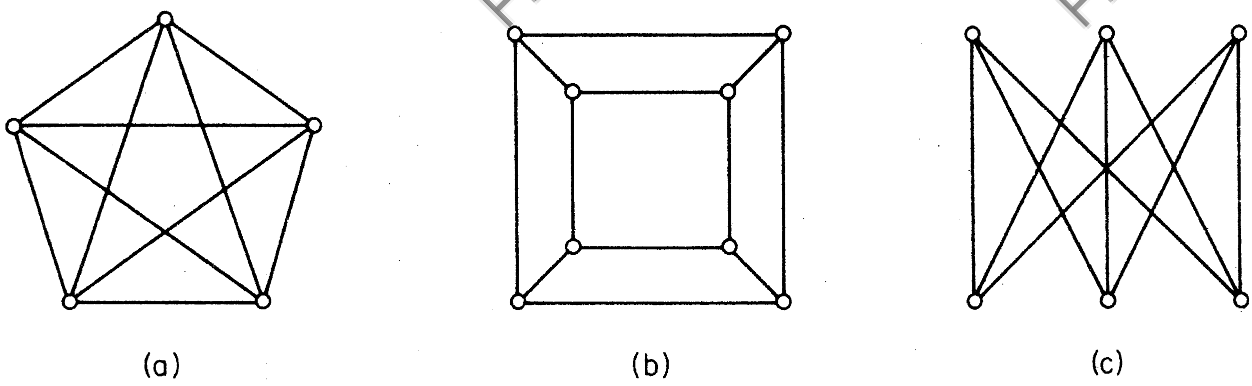


Figure 1.4. (*a*) K_5; (*b*) the cube; (*c*) $K_{3,3}$

graph is one whose vertex set can be partitioned into two subsets X and Y, so that each edge has one end in X and one end in Y; such a partition (X, Y) is called a *bipartition* of the graph. A *complete bipartite graph* is a simple bipartite graph with bipartition (X, Y) in which each vertex of X is joined to each vertex of Y; if $|X| = m$ and $|Y| = n$, such a graph is denoted by $K_{m,n}$. The graph defined by the vertices and edges of a cube (figure 1.4*b*) is bipartite; the graph in figure 1.4*c* is the complete bipartite graph $K_{3,3}$.

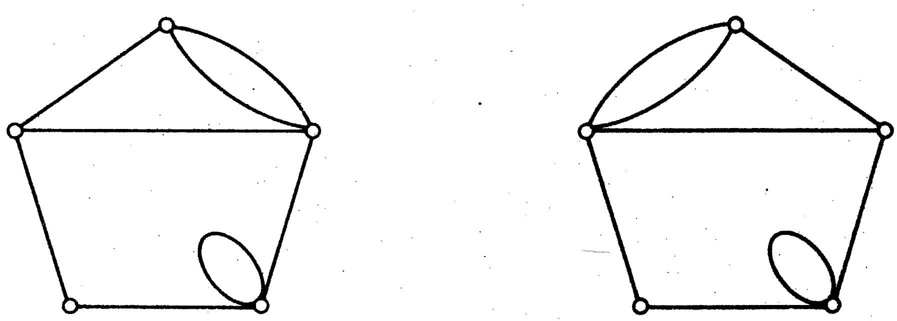
There are many other graphs whose structures are of special interest. Appendix III includes a selection of such graphs.

Exercises

1.2.1 Find an isomorphism between the graphs G and H of examples 1 and 2 different from the one given.

1.2.2 (*a*) Show that if $G \cong H$, then $\nu(G) = \nu(H)$ and $\varepsilon(G) = \varepsilon(H)$.
(*b*) Give an example to show that the converse is false.

1.2.3 Show that the following graphs are not isomorphic:

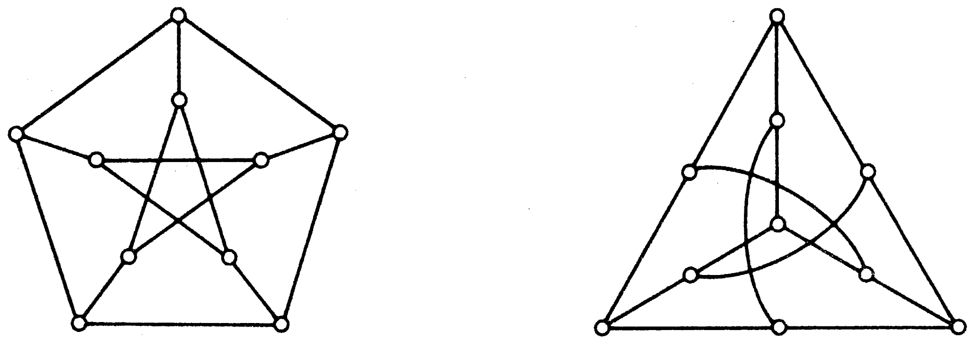


1.2.4 Show that there are eleven nonisomorphic simple graphs on four vertices.

1.2.5 Show that two simple graphs G and H are isomorphic if and only if there is a bijection $\theta : V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $\theta(u)\theta(v) \in E(H)$.

5585

1.2.6 Show that the following graphs are isomorphic:



1.2.7 Let G be simple. Show that $\varepsilon = \binom{\nu}{2}$ if and only if G is complete.

1.2.8 Show that

(*a*) $\varepsilon(K_{m,n}) = mn$;

(*b*) if G is simple and bipartite, then $\varepsilon \le \nu^2/4$.

1.2.9 A *k-partite graph* is one whose vertex set can be partitioned into k subsets so that no edge has both ends in any one subset; a *complete k-partite graph* is one that is simple and in which each vertex is joined to every vertex that is not in the same subset. The complete m-partite graph on n vertices in which each part has either $[n/m]$ or $\{n/m\}$ vertices is denoted by $T_{m,n}$. Show that

(*a*) $\varepsilon(T_{m,n}) = \binom{n-k}{2} + (m-1)\binom{k+1}{2}$, where $k = [n/m]$;

(*b*)* if G is a complete m-partite graph on n vertices, then $\varepsilon(G) \le \varepsilon(T_{m,n})$, with equality only if $G \cong T_{m,n}$.

1.2.10 The *k-cube* is the graph whose vertices are the ordered k-tuples of 0's and 1's, two vertices being joined if and only if they differ in exactly one coordinate. (The graph shown in figure 1.4*b* is just the 3-cube.) Show that the k-cube has 2^k vertices, $k2^{k-1}$ edges and is bipartite.

1.2.11 (*a*) The *complement* G^c of a simple graph G is the simple graph with vertex set V, two vertices being adjacent in G^c if and only if they are not adjacent in G. Describe the graphs K_n^c and $K_{m,n}^c$.

(*b*) A simple graph G is *self-complementary* if $G \cong G^c$. Show that if G is self-complementary, then $\nu \equiv 0, 1 \pmod 4$.

1.2.12 An *automorphism* of a graph is an isomorphism of the graph onto itself.

(*a*) Show, using exercise 1.2.5, that an automorphism of a simple graph G can be regarded as a permutation on V which preserves adjacency, and that the set of such permutations form a

FEUE WHEP

5586

group $\Gamma(G)$ (the *automorphism group* of G) under the usual operation of composition.

(*b*) Find $\Gamma(K_n)$ and $\Gamma(K_{m,n})$.

(*c*) Find a nontrivial simple graph whose automorphism group is the identity.

(*d*) Show that for any simple graph G, $\Gamma(G)=\Gamma(G^c)$.

(*e*) Consider the permutation group Λ with elements $(1)(2)(3)$, $(1,2,3)$ and $(1,3,2)$. Show that there is no simple graph G with vertex set $\{1,2,3\}$ such that $\Gamma(G)=\Lambda$.

(*f*) Find a simple graph G such that $\Gamma(G)\cong\Lambda$. (Frucht, 1939 has shown that every abstract group is isomorphic to the automorphism group of some graph.)

1.2.13 A simple graph G is *vertex-transitive* if, for any two vertices u and v, there is an element g in $\Gamma(G)$ such that $g(u)=g(v)$; G is *edge-transitive* if, for any two edges u_1v_1 and u_2v_2, there is an element h in $\Gamma(G)$ such that $h(\{u_1, v_1\})=\{u_2, v_2\}$. Find

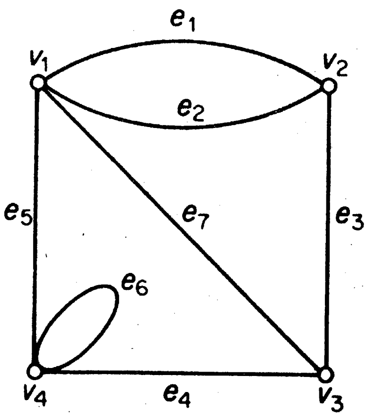
(*a*) a graph which is vertex-transitive but not edge-transitive;

(*b*) a graph which is edge-transitive but not vertex-transitive.

1.3 THE INCIDENCE AND ADJACENCY MATRICES

To any graph G there corresponds a $\nu\times\varepsilon$ matrix called the incidence matrix of G. Let us denote the vertices of G by $v_1, v_2, \ldots, v_\nu$ and the edges by $e_1, e_2, \ldots, e_\varepsilon$. Then the *incidence matrix* of G is the matrix $\mathbf{M}(G)=[m_{ij}]$, where m_{ij} is the number of times (0, 1 or 2) that v_i and e_j are incident. The incidence matrix of a graph is just a different way of specifying the graph.

Another matrix associated with G is the *adjacency matrix;* this is the $\nu\times\nu$ matrix $\mathbf{A}(G)=[a_{ij}]$, in which a_{ij} is the number of edges joining v_i and v_j. A graph, its incidence matrix, and its adjacency matrix are shown in figure 1.5.



G

$$\begin{array}{c|ccccccc}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\
\hline
v_1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
v_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
v_3 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
v_4 & 0 & 0 & 0 & 1 & 1 & 2 & 0
\end{array}$$

$\mathbf{M}(G)$

$$\begin{array}{c|cccc}
 & v_1 & v_2 & v_3 & v_4 \\
\hline
v_1 & 0 & 2 & 1 & 1 \\
v_2 & 2 & 0 & 1 & 0 \\
v_3 & 1 & 1 & 0 & 1 \\
v_4 & 1 & 0 & 1 & 1
\end{array}$$

$\mathbf{A}(G)$

Figure 1.5

FEUE WHEP

5587

The adjacency matrix of a graph is generally considerably smaller than its incidence matrix, and it is in this form that graphs are commonly stored in computers.

Exercises

1.3.1 Let **M** be the incidence matrix and **A** the adjacency matrix of a graph G.

(*a*) Show that every column sum of **M** is 2.
(*b*) What are the column sums of **A**?

1.3.2 Let G be bipartite. Show that the vertices of G can be enumerated so that the adjacency matrix of G has the form

$$\left[\begin{array}{c|c} \mathbf{0} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{0} \end{array}\right]$$

where $\mathbf{A}_{21}$ is the transpose of $\mathbf{A}_{12}$.

1.3.3* Show that if G is simple and the eigenvalues of **A** are distinct, then the automorphism group of G is abelian

1.4 SUBGRAPHS

A graph H is a *subgraph* of G (written $H \subseteq G$) if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, and ψ_H is the restriction of ψ_G to $E(H)$. When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a *proper subgraph* of G. If H is a subgraph of G, G is a *supergraph* of H. A *spanning subgraph* (or *spanning supergraph*) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

By deleting from G all loops and, for every pair of adjacent vertices, all but one link joining them, we obtain a simple spanning subgraph of G, called the *underlying simple graph* of G. Figure 1.6 shows a graph and its underlying simple graph.

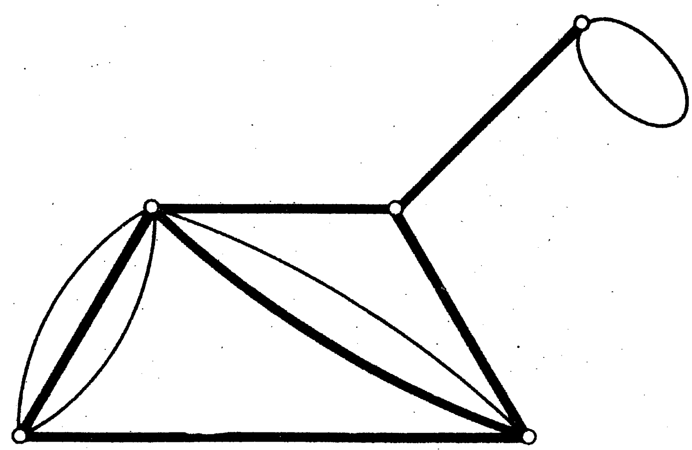


Figure 1.6. A graph and its underlying simple graph

FEUE WHEP

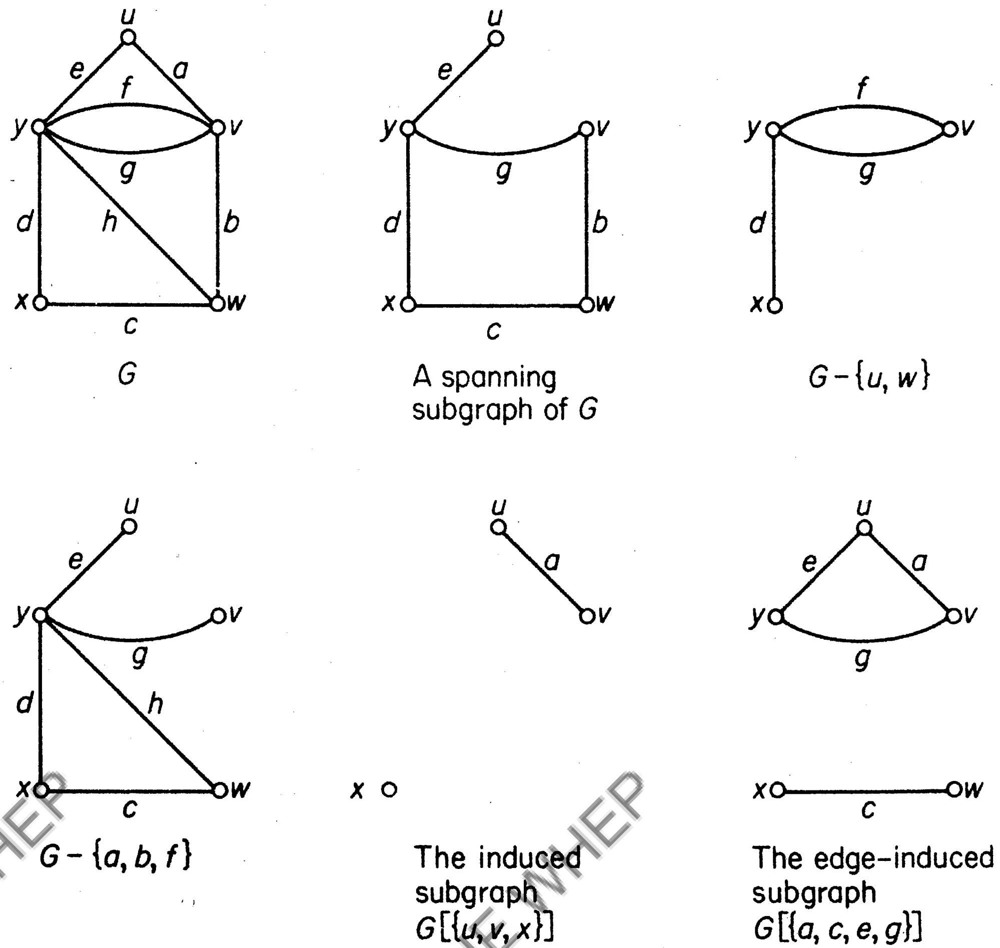


Figure 1.7

Suppose that V' is a nonempty subset of V. The subgraph of G whose vertex set is V' and whose edge set is the set of those edges of G that have both ends in V' is called the subgraph of G *induced* by V' and is denoted by $G[V']$; we say that $G[V']$ is an *induced subgraph* of G. The induced subgraph $G[V \backslash V']$ is denoted by $G - V'$; it is the subgraph obtained from G by deleting the vertices in V' together with their incident edges. If $V' = \{v\}$ we write $G - v$ for $G - \{v\}$.

Now suppose that E' is a nonempty subset of E. The subgraph of G whose vertex set is the set of ends of edges in E' and whose edge set is E' is called the subgraph of G *induced* by E' and is denoted by $G[E']$; $G[E']$ is an *edge-induced subgraph* of G. The spanning subgraph of G with edge set $E \backslash E'$ is written simply as $G - E'$; it is the subgraph obtained from G by deleting the edges in E'. Similarly, the graph obtained from G by adding a set of edges E' is denoted by $G + E'$. If $E' = \{e\}$ we write $G - e$ and $G + e$ instead of $G - \{e\}$ and $G + \{e\}$.

Subgraphs of these various types are depicted in figure 1.7.

Let G_1 and G_2 be subgraphs of G. We say that G_1 and G_2 are *disjoint* if they have no vertex in common, and *edge-disjoint* if they have no edge in common. The *union* $G_1 \cup G_2$ of G_1 and G_2 is the subgraph with vertex set

FEUE WHEP

5589

$V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$; if G_1 and G_2 are disjoint, we sometimes denote their union by $G_1 + G_2$. The *intersection* $G_1 \cap G_2$ of G_1 and G_2 is defined similarly, but in this case G_1 and G_2 must have at least one vertex in common.

Exercises

1.4.1 Show that every simple graph on n vertices is isomorphic to a subgraph of K_n.

1.4.2 Show that

(*a*) every induced subgraph of a complete graph is complete;
(*b*) every subgraph of a bipartite graph is bipartite.

1.4.3 Describe how $\mathbf{M}(G-E')$ and $\mathbf{M}(G-V')$ can be obtained from $\mathbf{M}(G)$, and how $\mathbf{A}(G-V')$ can be obtained from $\mathbf{A}(G)$.

1.4.4 Find a bipartite graph that is not isomorphic to a subgraph of any k-cube.

1.4.5* Let G be simple and let n be an integer with $1 < n < \nu - 1$. Show that if $\nu \geq 4$ and all induced subgraphs of G on n vertices have the same number of edges, then either $G \cong K_\nu$ or $G \cong K_\nu^c$.

1.5 VERTEX DEGREES

The *degree* $d_G(v)$ of a vertex v in G is the number of edges of G incident with v, each loop counting as two edges. We denote by $\delta(G)$ and $\Delta(G)$ the minimum and maximum degrees, respectively, of vertices of G.

Theorem 1.1

$$\sum_{v \in V} d(v) = 2\varepsilon$$

Proof Consider the incidence matrix $\mathbf{M}$. The sum of the entries in the row corresponding to vertex v is precisely $d(v)$, and therefore $\sum_{v \in V} d(v)$ is just the sum of all entries in $\mathbf{M}$. But this sum is also 2ε, since (exercise 1.3.1*a*) each of the ε column sums of $\mathbf{M}$ is 2 □

Corollary 1.1 In any graph, the number of vertices of odd degree is even.

Proof Let V_1 and V_2 be the sets of vertices of odd and even degree in G, respectively. Then

$$\sum_{v \in V_1} d(v) + \sum_{v \in V_2} d(v) = \sum_{v \in V} d(v)$$

is even, by theorem 1.1. Since $\sum_{v \in V_2} d(v)$ is also even, it follows that $\sum_{v \in V_1} d(v)$ is even. Thus $|V_1|$ is even □

FEUE WHEP

5590

A graph G is *k-regular* if $d(v)=k$ for all $v \in V$; a *regular graph* is one that is k-regular for some k. Complete graphs and complete bipartite graphs $K_{n,n}$ are regular; so, also, are the k-cubes.

Exercises

1.5.1 Show that $\delta \leq 2\varepsilon/\nu \leq \Delta$.

1.5.2 Show that if G is simple, the entries on the diagonals of both $\mathbf{MM}'$ and $\mathbf{A}^2$ are the degrees of the vertices of G.

1.5.3 Show that if a k-regular bipartite graph with $k>0$ has bipartition (X, Y), then $|X|=|Y|$.

1.5.4 Show that, in any group of two or more people, there are always two with exactly the same number of friends inside the group.

1.5.5 If G has vertices $v_1, v_2, \ldots, v_\nu$, the sequence $(d(v_1), d(v_2), \ldots, d(v_\nu))$ is called a *degree sequence* of G. Show that a sequence $(d_1, d_2, \ldots, d_n)$ of non-negative integers is a degree sequence of some graph if and only if $\sum_{i=1}^{n} d_i$ is even.

1.5.6 A sequence $\mathbf{d}=(d_1, d_2, \ldots, d_n)$ is *graphic* if there is a simple graph with degree sequence $\mathbf{d}$. Show that

(*a*) the sequences (7, 6, 5, 4, 3, 3, 2) and (6, 6, 5, 4, 3, 3, 1) are not graphic;

(*b*) if $\mathbf{d}$ is graphic and $d_1 \geq d_2 \geq \ldots \geq d_n$, then $\sum_{i=1}^{n} d_i$ is even and

$$\sum_{i=1}^{k} d_i \leq k(k-1)+\sum_{i=k+1}^{n} \min\{k, d_i\} \quad \text{for} \quad 1 \leq k \leq n$$

(Erdös and Gallai, 1960 have shown that this necessary condition is also sufficient for $\mathbf{d}$ to be graphic.)

1.5.7 Let $\mathbf{d}=(d_1, d_2, \ldots, d_n)$ be a nonincreasing sequence of non-negative integers, and denote the sequence $(d_2-1, d_3-1, \ldots, d_{d_1+1}-1, d_{d_1+2}, \ldots, d_n)$ by $\mathbf{d}'$.

(*a*)* Show that $\mathbf{d}$ is graphic if and only if $\mathbf{d}'$ is graphic.

(*b*) Using (*a*), describe an algorithm for constructing a simple graph with degree sequence $\mathbf{d}$, if such a graph exists.

(V. Havel, S. Hakimi)

1.5.8* Show that a loopless graph G contains a bipartite spanning subgraph H such that $d_H(v) \geq \frac{1}{2}d_G(v)$ for all $v \in V$.

1.5.9* Let $S=\{x_1, x_2, \ldots, x_n\}$ be a set of points in the plane such that the distance between any two points is at least one. Show that there are at most $3n$ pairs of points at distance exactly one.

1.5.10 The *edge graph* of a graph G is the graph with vertex set $E(G)$ in which two vertices are joined if and only if they are adjacent edges in

5591

G. Show that, if G is simple

(*a*) the edge graph of G has $\varepsilon(G)$ vertices and $\sum_{v\in V(G)} \binom{d_G(v)}{2}$ edges;

(*b*) the edge graph of K_5 is isomorphic to the complement of the graph featured in exercise 1.2.6.

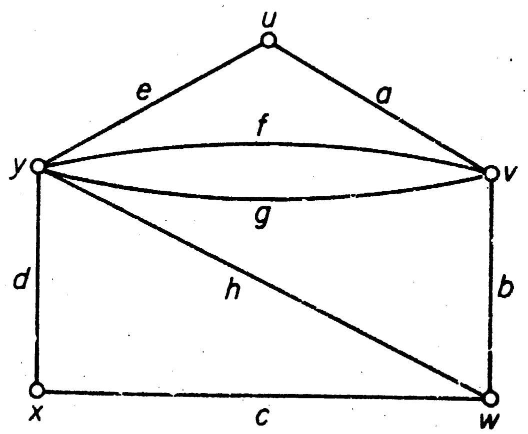
1.6 PATHS AND CONNECTION

A *walk* in G is a finite non-null sequence $W = v_0e_1v_1e_2v_2 \ldots e_kv_k$, whose terms are alternately vertices and edges, such that, for $1 \le i \le k$, the ends of e_i are v_{i-1} and v_i. We say that W is a walk *from* v_0 *to* v_k, or a (v_0, v_k)*-walk*. The vertices v_0 and v_k are called the *origin* and *terminus* of W, respectively, and $v_1, v_2, \ldots, v_{k-1}$ its *internal vertices*. The integer k is the *length* of W.

If $W = v_0e_1v_1 \ldots e_kv_k$ and $W' = v_ke_{k+1}v_{k+1} \ldots e_lv_l$ are walks, the walk $v_ke_kv_{k-1} \ldots e_1v_0$, obtained by reversing W, is denoted by W^{-1} and the walk $v_0e_1v_1 \ldots e_lv_l$, obtained by concatenating W and W' at v_k, is denoted by WW'. A *section* of a walk $W = v_0e_1v_1 \ldots e_kv_k$ is a walk that is a subsequence $v_ie_{i+1}v_{i+1} \ldots e_jv_j$ of consecutive terms of W; we refer to this subsequence as the (v_i, v_j)*-section* of W.

In a simple graph, a walk $v_0e_1v_1 \ldots e_kv_k$ is determined by the sequence $v_0v_1 \ldots v_k$ of its vertices; hence a walk in a simple graph can be specified simply by its vertex sequence. Moreover, even in graphs that are not simple, we shall sometimes refer to a sequence of vertices in which consecutive terms are adjacent as a 'walk'. In such cases it should be understood that the discussion is valid for every walk with that vertex sequence.

If the edges $e_1, e_2, \ldots, e_k$ of a walk W are distinct, W is called a *trail;* in this case the length of W is just $\varepsilon(W)$. If, in addition, the vertices $v_0, v_1, \ldots, v_k$ are distinct, W is called a *path*. Figure 1.8 illustrates a walk, a trail and a path in a graph. We shall also use the word 'path' to denote a graph or subgraph whose vertices and edges are the terms of a path.

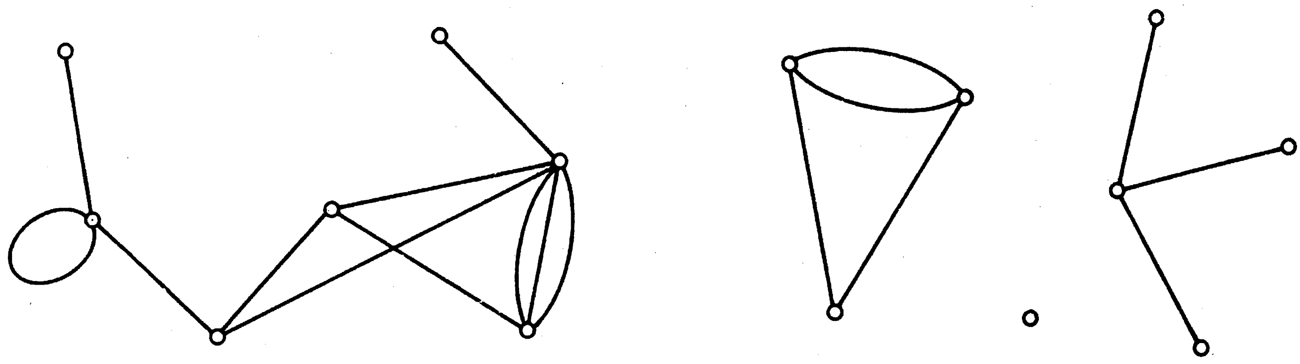


Walk: *uavfyfvgyhwbv*
Trail: *wcxdyhwbvgy*
Path: *xcwhyeuav*

Figure 1.8

FEUE WHEP

5592



(a)

(b)

Figure 1.9. (*a*) A connected graph; (*b*) a disconnected graph with three components

Two vertices u and v of G are said to be *connected* if there is a (u, v)-path in G. Connection is an equivalence relation on the vertex set V. Thus there is a partition of V into nonempty subsets $V_1, V_2, \ldots, V_\omega$ such that two vertices u and v are connected if and only if both u and v belong to the same set V_i. The subgraphs $G[V_1], G[V_2], \ldots, G[V_\omega]$ are called the *components* of G. If G has exactly one component, G is *connected;* otherwise G is *disconnected.* We denote the number of components of G by $\omega(G)$. Connected and disconnected graphs are depicted in figure 1.9.

Exercises

1.6.1 Show that if there is a (u, v)-walk in G, then there is also a (u, v)-path in G.

1.6.2 Show that the number of (v_i, v_j)-walks of length k in G is the (i, j)th entry of $\mathbf{A}^k$.

1.6.3 Show that if G is simple and $\delta \geq k$, then G has a path of length k.

1.6.4 Show that G is connected if and only if, for every partition of V into two nonempty sets V_1 and V_2, there is an edge with one end in V_1 and one end in V_2.

1.6.5 (*a*) Show that if G is simple and $\varepsilon > \binom{\nu - 1}{2}$, then G is connected.

(*b*) For $\nu > 1$, find a disconnected simple graph G with $\varepsilon = \binom{\nu - 1}{2}$.

1.6.6 (*a*) Show that if G is simple and $\delta > [\nu/2] - 1$, then G is connected.

(*b*) Find a disconnected $([\nu/2] - 1)$-regular simple graph for ν even.

1.6.7 Show that if G is disconnected, then G^c is connected.

1.6.8 (*a*) Show that if $e \in E$, then $\omega(G) \leq \omega(G - e) \leq \omega(G) + 1$.

(*b*) Let $v \in V$. Show that $G - e$ cannot, in general, be replaced by $G - v$ in the above inequality.

1.6.9 Show that if G is connected and each degree in G is even, then, for any $v \in V$, $\omega(G - v) \leq \frac{1}{2}d(v)$.

FEUE WHEP

5593

1.6.10 Show that any two longest paths in a connected graph have a vertex in common.

1.6.11 If vertices u and v are connected in G, the *distance* between u and v in G, denoted by $d_G(u, v)$, is the length of a shortest (u, v)-path in G; if there is no path connecting u and v we define $d_G(u, v)$ to be infinite. Show that, for any three vertices u, v and w, $d(u, v)+d(v, w)\geq d(u, w)$.

1.6.12 The *diameter* of G is the maximum distance between two vertices of G. Show that if G has diameter greater than three, then G^c has diameter less than three.

1.6.13 Show that if G is simple with diameter two and $\Delta=\nu-2$, then $\varepsilon\geq 2\nu-4$.

1.6.14 Show that if G is simple and connected but not complete, then G has three vertices u, v and w such that $uv, vw\in E$ and $uw\notin E$.

1.7 CYCLES

A walk is *closed* if it has positive length and its origin and terminus are the same. A closed trail whose origin and internal vertices are distinct is a *cycle*. Just as with paths we sometimes use the term 'cycle' to denote a graph corresponding to a cycle. A cycle of length k is called a k*-cycle;* a k-cycle is *odd* or *even* according as k is odd or even. A 3-cycle is often called a *triangle*. Examples of a closed trail and a cycle are given in figure 1.10.

Using the concept of a cycle, we can now present a characterisation of bipartite graphs.

Theorem 1.2 A graph is bipartite if and only if it contains no odd cycle.

Proof Suppose that G is bipartite with bipartition (X, Y), and let $C=v_0v_1\ldots v_kv_0$ be a cycle of G. Without loss of generality we may assume that $v_0\in X$. Then, since $v_0v_1\in E$ and G is bipartite, $v_1\in Y$. Similarly $v_2\in X$ and, in general, $v_{2i}\in X$ and $v_{2i+1}\in Y$. Since $v_0\in X$, $v_k\in Y$. Thus $k=2i+1$, for some i, and it follows that C is even.

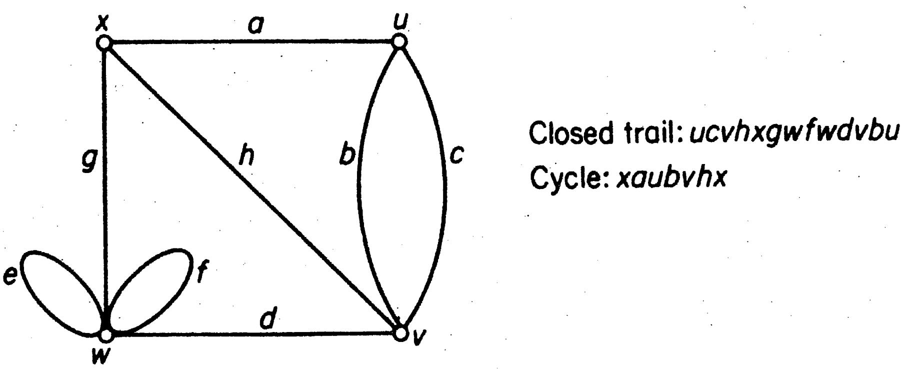


Figure 1.10

FEUE WHEP

FÊUE WHEP

It clearly suffices to prove the converse for connected graphs. Let G be a connected graph that contains no odd cycles. We choose an arbitrary vertex u and define a partition (X, Y) of V by setting

$$X = \{x \in V \mid d(u, x) \quad \text{is even}\}$$
$$Y = \{y \in V \mid d(u, y) \quad \text{is odd}\}$$

We shall show that (X, Y) is a bipartition of G. Suppose that v and w are two vertices of X. Let P be a shortest (u, v)-path and Q be a shortest (u, w)-path. Denote by u_1 the last vertex common to P and Q. Since P and Q are shortest paths, the (u, u_1)-sections of both P and Q are shortest (u, u_1)-paths and, therefore, have the same length. Now, since the lengths of both P and Q are even, the lengths of the (u_1, v)-section P_1 of P and the (u_1, w)-section Q_1 of Q must have the same parity. It follows that the (v, w)-path $P_1^{-1}Q_1$ is of even length. If v were joined to w, $P_1^{-1}Q_1wv$ would be a cycle of odd length, contrary to the hypothesis. Therefore no two vertices in X are adjacent; similarly, no two vertices in Y are adjacent □

Exercises

1.7.1 Show that if an edge e is in a closed trail of G, then e is in a cycle of G.

1.7.2 Show that if $\delta \geq 2$, then G contains a cycle.

1.7.3* Show that if G is simple and $\delta \geq 2$, then G contains a cycle of length at least $\delta + 1$.

1.7.4 The *girth* of G is the length of a shortest cycle in G; if G has no cycles we define the girth of G to be infinite. Show that

(*a*) a k-regular graph of girth four has at least $2k$ vertices, and (up to isomorphism) there exists exactly one such graph on $2k$ vertices;

(*b*) a k-regular graph of girth five has at least $k^2 + 1$ vertices.

1.7.5 Show that a k-regular graph of girth five and diameter two has exactly $k^2 + 1$ vertices, and find such a graph for $k = 2, 3$. (Hoffman and Singleton, 1960 have shown that such a graph can exist only if $k = 2, 3, 7$ and, possibly, 57.)

1.7.6 Show that

(*a*) if $\varepsilon \geq \nu$, G contains a cycle;

(*b*)* if $\varepsilon \geq \nu + 4$, G contains two edge-disjoint cycles. (L. Pósa)

APPLICATIONS

1.8 THE SHORTEST PATH PROBLEM

With each edge e of G let there be associated a real number $w(e)$, called its *weight*. Then G, together with these weights on its edges, is called a *weighted*

5595

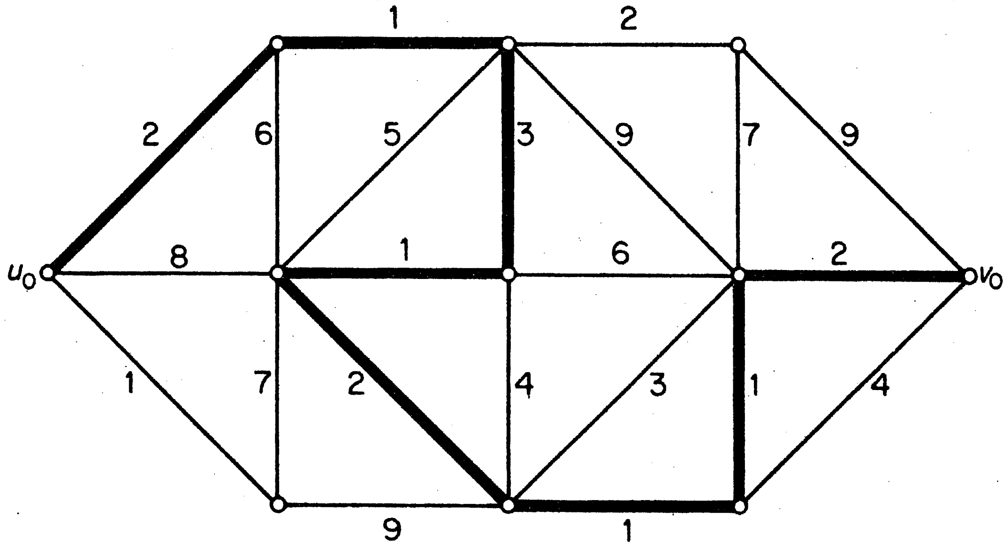


Figure 1.11. A (u_0, v_0)-path of minimum weight

graph. Weighted graphs occur frequently in applications of graph theory. In the friendship graph, for example, weights might indicate intensity of friendship; in the communications graph, they could represent the construction or maintenance costs of the various communication links.

If H is a subgraph of a weighted graph, the *weight* $w(H)$ of H is the sum of the weights $\sum_{e\in E(H)} w(e)$ on its edges. Many optimisation problems amount to finding, in a weighted graph, a subgraph of a certain type with minimum (or maximum) weight. One such is the *shortest path problem:* given a railway network connecting various towns, determine a shortest route between two specified towns in the network.

Here one must find, in a weighted graph, a path of minimum weight connecting two specified vertices u_0 and v_0; the weights represent distances by rail between directly-linked towns, and are therefore non-negative. The path indicated in the graph of figure 1.11 is a (u_0, v_0)-path of minimum weight (exercise 1.8.1).

We now present an algorithm for solving the shortest path problem. For clarity of exposition, we shall refer to the weight of a path in a weighted graph as its *length*; similarly the minimum weight of a (u, v)-path will be called the *distance* between u and v and denoted by $d(u, v)$. These definitions coincide with the usual notions of length and distance, as defined in section 1.6, when all the weights are equal to one.

It clearly suffices to deal with the shortest path problem for simple graphs; so we shall assume here that G is simple. We shall also assume that all the weights are positive. This, again, is not a serious restriction because, if the weight of an edge is zero, then its ends can be identified. We adopt the convention that $w(uv) = \infty$ if $uv \notin E$.

FÊUE WHEP

5596

The algorithm to be described was discovered by Dijkstra (1959) and, independently, by Whiting and Hillier (1960). It finds not only a shortest (u_0, v_0)-path, but shortest paths from u_0 to all other vertices of G. The basic idea is as follows.

Suppose that S is a proper subset of V such that $u_0 \in S$, and let $\bar{S}$ denote $V \backslash S$. If $P = u_0 \ldots \bar{u}\bar{v}$ is a shortest path from u_0 to $\bar{S}$ then clearly $\bar{u} \in S$ and the $(u_0, \bar{u})$-section of P must be a shortest $(u_0, \bar{u})$-path. Therefore

$$d(u_0, \bar{v}) = d(u_0, \bar{u}) + w(\bar{u}\bar{v})$$

and the distance from u_0 to $\bar{S}$ is given by the formula

$$d(u_0, \bar{S}) = \min_{\substack{u \in S \\ v \in \bar{S}}} \{d(u_0, u) + w(uv)\} \tag{1.1}$$

This formula is the basis of Dijkstra's algorithm. Starting with the set $S_0 = \{u_0\}$, an increasing sequence $S_0, S_1, \ldots, S_{\nu-1}$ of subsets of V is constructed, in such a way that, at the end of stage i, shortest paths from u_0 to all vertices in S_i are known.

The first step is to determine a vertex nearest to u_0. This is achieved by computing $d(u_0, \bar{S}_0)$ and selecting a vertex $u_1 \in \bar{S}_0$ such that $d(u_0, u_1) = d(u_0, \bar{S}_0)$; by (1.1)

$$d(u_0, \bar{S}_0) = \min_{\substack{u \in S_0 \\ v \in \bar{S}_0}} \{d(u_0, u) + w(uv)\} = \min_{v \in \bar{S}_0} \{w(u_0 v)\}$$

and so $d(u_0, \bar{S}_0)$ is easily computed. We now set $S_1 = \{u_0, u_1\}$ and let P_1 denote the path u_0u_1; this is clearly a shortest (u_0, u_1)-path. In general, if the set $S_k = \{u_0, u_1, \ldots, u_k\}$ and corresponding shortest paths $P_1, P_2, \ldots, P_k$ have already been determined, we compute $d(u_0, \bar{S}_k)$ using (1.1) and select a vertex $u_{k+1} \in \bar{S}_k$ such that $d(u_0, u_{k+1}) = d(u_0, \bar{S}_k)$. By (1.1), $d(u_0, u_{k+1}) = d(u_0, u_j) + w(u_j u_{k+1})$ for some $j \leq k$; we get a shortest (u_0, u_{k+1})-path by adjoining the edge $u_j u_{k+1}$ to the path P_j.

We illustrate this procedure by considering the weighted graph depicted in figure 1.12*a*. Shortest paths from u_0 to the remaining vertices are determined in seven stages. At each stage, the vertices to which shortest paths have been found are indicated by solid dots, and each is labelled by its distance from u_0; initially u_0 is labelled 0. The actual shortest paths are indicated by solid lines. Notice that, at each stage, these shortest paths together form a connected graph without cycles; such a graph is called a *tree*, and we can think of the algorithm as a 'tree-growing' procedure. The final tree, in figure 1.12*h*, has the property that, for each vertex v, the path connecting u_0 and v is a shortest (u_0, v)-path.

Dijkstra's algorithm is a refinement of the above procedure. This refinement is motivated by the consideration that, if the minimum in (1.1) were to be computed from scratch at each stage, many comparisons would be

5597

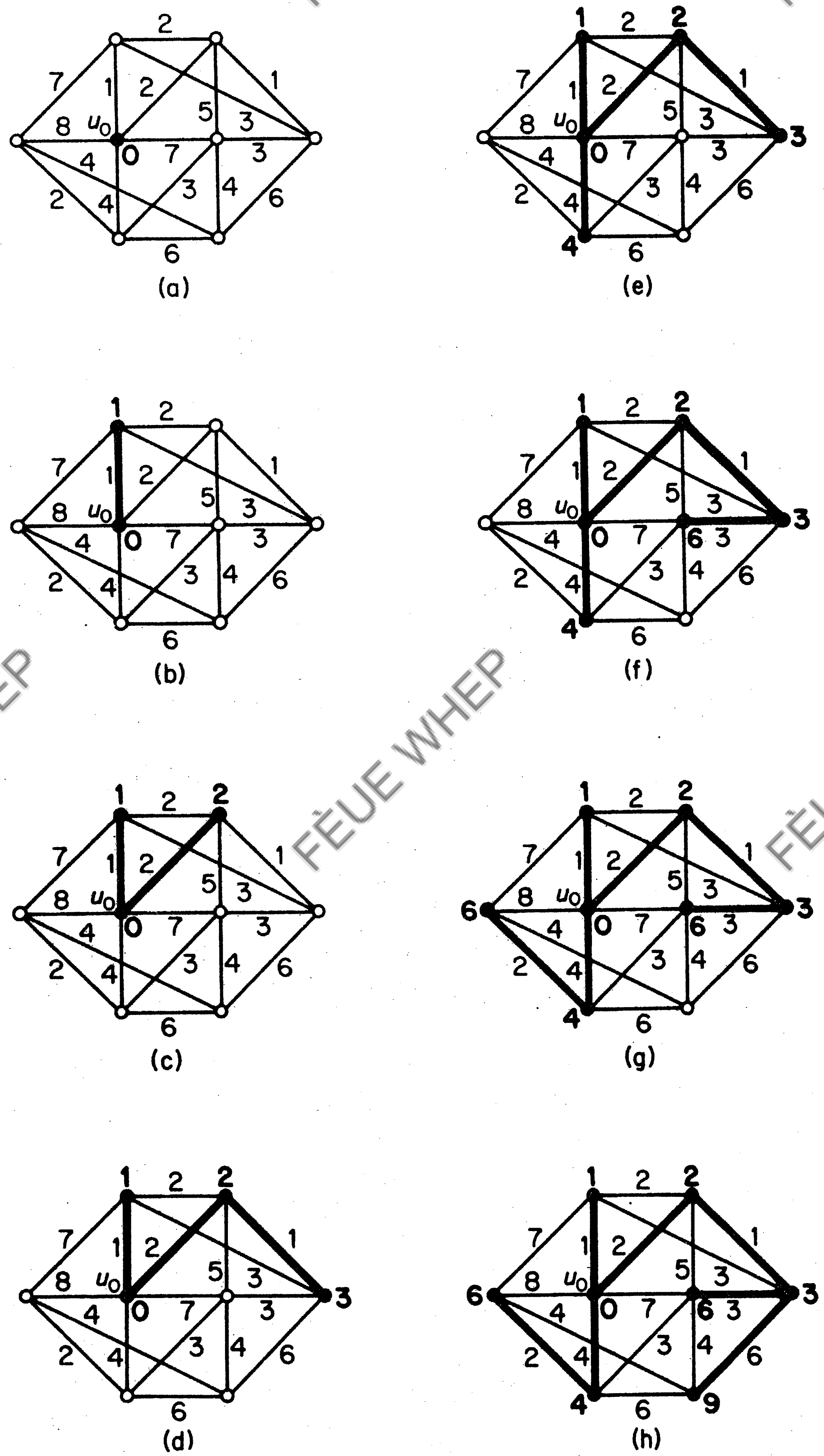


Figure 1.12. Shortest path algorithm

FEUE WHEP

5598

repeated unnecessarily. To avoid such repetitions, and to retain computational information from one stage to the next, we adopt the following labelling procedure. Throughout the algorithm, each vertex v carries a label $l(v)$ which is an upper bound on $d(u_0, v)$. Initially $l(u_0)=0$ and $l(v)=\infty$ for $v\neq u_0$. (In actual computations ∞ is replaced by any sufficiently large number.) As the algorithm proceeds, these labels are modified so that, at the end of stage i,

$$l(u)=d(u_0, u) \quad \text{for} \quad u\in S_i$$

and

$$l(v)=\min_{u\in S_{i-1}}\{d(u_0, u)+w(uv)\} \quad \text{for} \quad v\in \bar{S}_i$$

Dijkstra's Algorithm

1. Set $l(u_0)=0$, $l(v)=\infty$ for $v\neq u_0$, $S_0=\{u_0\}$ and $i=0$.
2. For each $v\in\bar{S}_i$, replace $l(v)$ by $\min\{l(v), l(u_i)+w(u_iv)\}$. Compute $\min_{v\in\bar{S}_i}\{l(v)\}$ and let u_{i+1} denote a vertex for which this minimum is attained. Set $S_{i+1}=S_i\cup\{u_{i+1}\}$.
3. If $i=\nu-1$, stop. If $i<\nu-1$, replace i by $i+1$ and go to step 2.

When the algorithm terminates, the distance from u_0 to v is given by the final value of the label $l(v)$. (If our interest is in determining the distance to one specific vertex v_0, we stop as soon as some u_j equals v_0.) A flow diagram summarising this algorithm is shown in figure 1.13.

As described above, Dijkstra's algorithm determines only the distances from u_0 to all the other vertices, and not the actual shortest paths. These shortest paths can, however, be easily determined by keeping track of the predecessors of vertices in the tree (exercise 1.8.2).

Dijkstra's algorithm is an example of what Edmonds (1965) calls a good algorithm. A graph-theoretic algorithm is *good* if the number of computational steps required for its implementation on any graph G is bounded above by a polynomial in ν and ε (such as $3\nu^2\varepsilon$). An algorithm whose implementation may require an exponential number of steps (such as 2^ν) might be very inefficient for some large graphs.

To see that Dijkstra's algorithm is good, note that the computations involved in boxes 2 and 3 of the flow diagram, totalled over all iterations, require $\nu(\nu-1)/2$ additions and $\nu(\nu-1)$ comparisons. One of the questions that is not elaborated upon in the flow diagram is the matter of deciding whether a vertex belongs to $\bar{S}$ or not (box 1). Dreyfus (1969) reports a technique for doing this that requires a total of $(\nu-1)^2$ comparisons. Hence, if we regard either a comparison or an addition as a basic computational unit, the total number of computations required for this algorithm is approximately $5\nu^2/2$, and thus of order ν^2. (A function $f(\nu, \varepsilon)$ is of *order*

5599

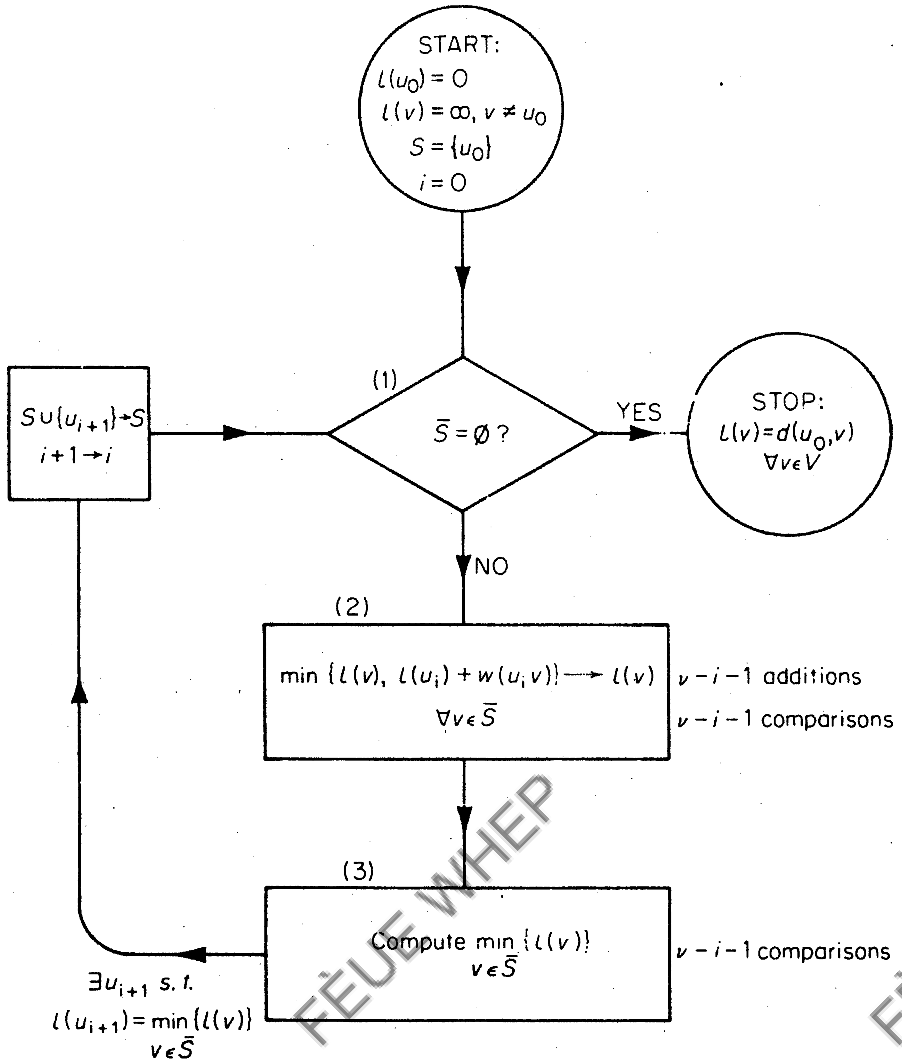


Figure 1.13. Dijkstra's algorithm

$g(\nu, \varepsilon)$ if there exists a positive constant c such that $f(\nu, \varepsilon)/g(\nu, \varepsilon) \leq c$ for all ν and ε.)

Although the shortest path problem can be solved by a good algorithm, there are many problems in graph theory for which no good algorithm is known. We refer the reader to Aho, Hopcroft and Ullman (1974) for further details.

Exercises

1.8.1 Find shortest paths from u_0 to all other vertices in the weighted graph of figure 1.11.

1.8.2 What additional instructions are needed in order that Dijkstra's algorithm determine shortest paths rather than merely distances?

1.8.3 A company has branches in each of six cities $C_1, C_2, \ldots, C_6$. The fare for a direct flight from C_i to C_j is given by the (i, j)th entry in the following matrix (∞ indicates that there is no direct flight):

5600

$$\begin{bmatrix} 0 & 50 & \infty & 40 & 25 & 10 \\ 50 & 0 & 15 & 20 & \infty & 25 \\ \infty & 15 & 0 & 10 & 20 & \infty \\ 40 & 20 & 10 & 0 & 10 & 25 \\ 25 & \infty & 20 & 10 & 0 & 55 \\ 10 & 25 & \infty & 25 & 55 & 0 \end{bmatrix}$$

The company is interested in computing a table of cheapest routes between pairs of cities. Prepare such a table.

1.8.4 A wolf, a goat and a cabbage are on one bank of a river. A ferryman wants to take them across, but, since his boat is small, he can take only one of them at a time. For obvious reasons, neither the wolf and the goat nor the goat and the cabbage can be left unguarded. How is the ferryman going to get them across the river?

1.8.5 Two men have a full eight-gallon jug of wine, and also two empty jugs of five and three gallons capacity, respectively. What is the simplest way for them to divide the wine equally?

1.8.6 Describe a good algorithm for determining

(*a*) the components of a graph;
(*b*) the girth of a graph.

How good are your algorithms?

1.9 SPERNER'S LEMMA

Every continuous mapping f of a closed n-disc to itself has a fixed point (that is, a point x such that $f(x)=x$). This powerful theorem, known as *Brouwer's fixed-point theorem*, has a wide range of applications in modern mathematics. Somewhat surprisingly, it is an easy consequence of a simple combinatorial lemma due to Sperner (1928). And, as we shall see in this section, Sperner's lemma is, in turn, an immediate consequence of corollary 1.1.

Sperner's lemma concerns the decomposition of a simplex (line segment, triangle, tetrahedron and so on) into smaller simplices. For the sake of simplicity we shall deal with the two-dimensional case.

Let T be a closed triangle in the plane. A subdivision of T into a finite number of smaller triangles is said to be *simplicial* if any two intersecting triangles have either a vertex or a whole side in common (see figure 1.14*a*).

Suppose that a simplicial subdivision of T is given. Then a labelling of the vertices of triangles in the subdivision in three symbols 0, 1 and 2 is said to be *proper* if

(i) the three vertices of T are labelled 0, 1 and 2 (in any order), and
(ii) for $0 \leq i < j \leq 2$, each vertex on the side of T joining vertices labelled i and j is labelled either i or j.

5601

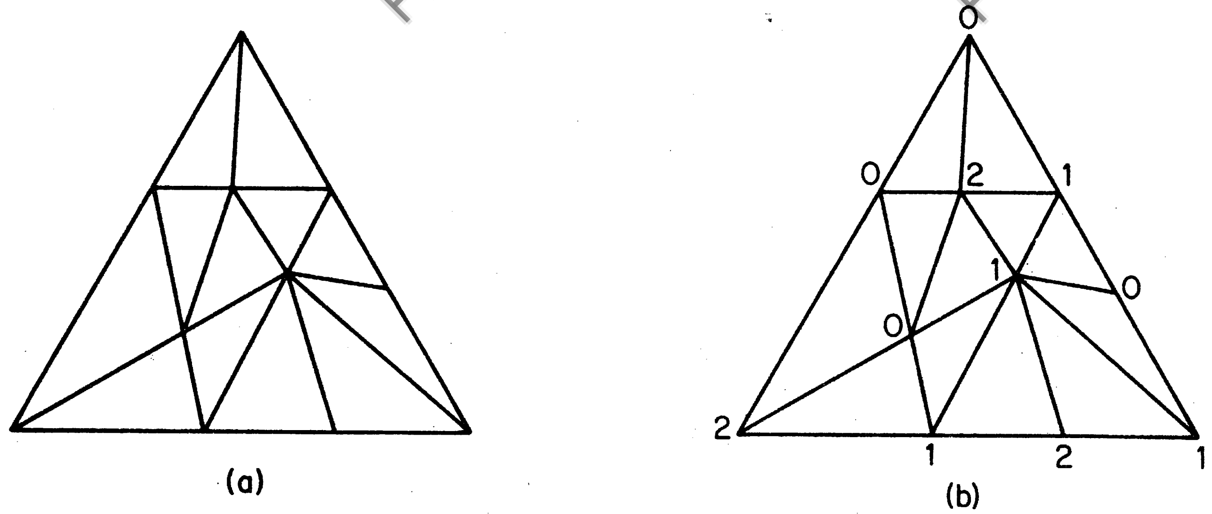


Figure 1.14. (*a*) A simplicial subdivision of a triangle; (*b*) a proper labelling of the subdivision

We call a triangle in the subdivision whose vertices receive all three labels a *distinguished* triangle. The proper labelling in figure 1.14*b* has three distinguished triangles.

Theorem 1.3 (*Sperner's lemma*) Every properly labelled simplicial subdivision of a triangle has an odd number of distinguished triangles.

Proof Let T_0 denote the region outside T, and let $T_1, T_2, \ldots, T_n$ be the triangles of the subdivision. Construct a graph on the vertex set $\{v_0, v_1, \ldots, v_n\}$ by joining v_i and v_j whenever the common boundary of T_i and T_j is an edge with labels 0 and 1 (see figure 1.15).

In this graph, v_0 is clearly of odd degree (exercise 1.9.1). It follows from corollary 1.1 that an odd number of the vertices $v_1, v_2, \ldots, v_n$ are of odd degree. Now it is easily seen that none of these vertices can have degree

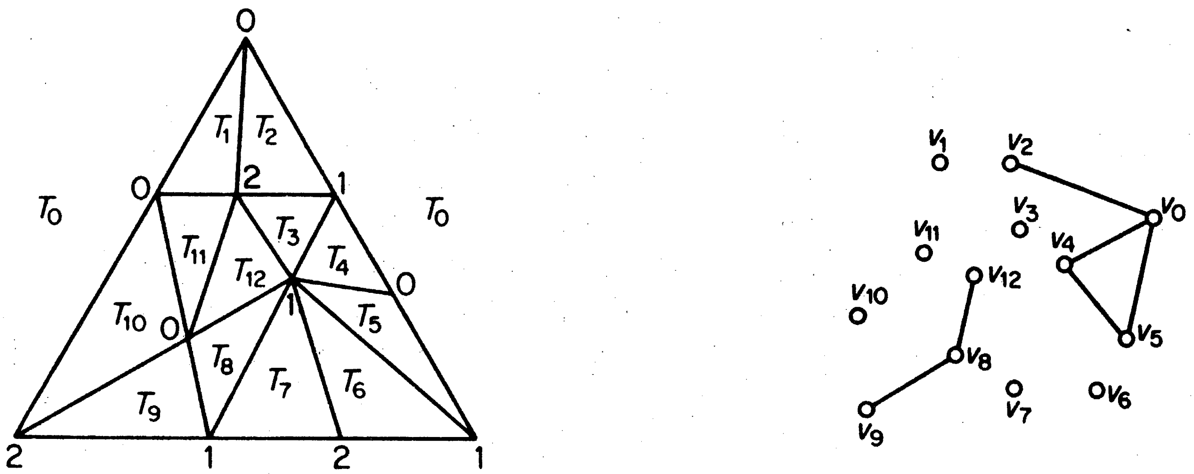


Figure 1.15

FEUE WHEP

5602

three, and so those with odd degree must have degree one. But a vertex v_i is of degree one if and only if the triangle T_i is distinguished □

We shall now briefly indicate how Sperner's lemma can be used to deduce Brouwer's fixed-point theorem. Again, for simplicity, we shall only deal with the two-dimensional case. Since a closed 2-disc is homeomorphic to a closed triangle, it suffices to prove that a continuous mapping of a closed triangle to itself has a fixed point.

Let T be a given closed triangle with vertices x_0, x_1 and x_2. Then each point x of T can be written uniquely as $x = a_0x_0 + a_1x_1 + a_2x_2$, where each $a_i \geq 0$ and $\Sigma\, a_i = 1$, and we can represent x by the vector (a_0, a_1, a_2); the real numbers a_0, a_1 and a_2 are called the *barycentric coordinates* of x.

Now let f be any continuous mapping of T to itself, and suppose that

$$f(a_0, a_1, a_2) = (a_0', a_1', a_2')$$

Define S_i as the set of points (a_0, a_1, a_2) in T for which $a_i' \leq a_i$. To show that f has a fixed point, it is enough to show that $S_0 \cap S_1 \cap S_2 \neq \emptyset$. For suppose that $(a_0, a_1, a_2) \in S_0 \cap S_1 \cap S_2$. Then, by the definition of S_i, we have that $a_i' \leq a_i$ for each i, and this, coupled with the fact that $\Sigma\, a_i' = \Sigma\, a_i$, yields

$$(a_0', a_1', a_2') = (a_0, a_1, a_2)$$

In other words, (a_0, a_1, a_2) is a fixed point of f.

So consider an arbitrary subdivision of T and a proper labelling such that each vertex labelled i belongs to S_i; the existence of such a labelling is easily seen (exercise 1.9.2*a*). It follows from Sperner's lemma that there is a triangle in the subdivision whose three vertices belong to S_0, S_1 and S_2. Now this holds for any subdivision of T and, since it is possible to choose subdivisions in which each of the smaller triangles are of arbitrarily small diameter, we conclude that there exist three points of S_0, S_1 and S_2 which are arbitrarily close to one another. Because the sets S_i are closed (exercise 1.9.2*b*), one may deduce that $S_0 \cap S_1 \cap S_2 \neq \emptyset$.

For details of the above proof and other applications of Sperner's lemma, the reader is referred to Tompkins (1964).

Exercises

1.9.1 In the proof of Sperner's lemma, show that the vertex v_0 is of odd degree.

1.9.2 In the proof of Brouwer's fixed-point theorem, show that
- (*a*) there exists a proper labelling such that each vertex labelled i belongs to S_i;
- (*b*) the sets S_i are closed.

1.9.3 State and prove Sperner's lemma for higher dimensional simplices.

5603

REFERENCES

Aho, A. V., Hopcroft, J. E. and Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numer. Math.*, **1,** 269–71

Dreyfus, S. E. (1969). An appraisal of some shortest-path algorithms. *Operations Res.*, **17,** 395–412

Edmonds, J. (1965). Paths, trees and flowers. *Canad. J. Math.*, **17,** 449–67

Erdös, P. and Gallai, T. (1960). Graphs with prescribed degrees of vertices (Hungarian). *Mat. Lapok*, **11,** 264–74

Frucht, R. (1939). Herstellung von Graphen mit vorgegebener abstrakter Gruppe. *Compositio Math.*, **6,** 239–50

Hoffman, A. J. and Singleton, R. R. (1960). On Moore graphs with diameters 2 and 3. *IBM J. Res. Develop.*, **4,** 497–504

Sperner, E. (1928). Neuer Beweis für die Invarianz der Dimensionszahl und des Gebietes. *Hamburger Abhand.*, **6,** 265–72

Tompkins, C. B. (1964). Sperner's lemma and some extensions, in *Applied Combinatorial Mathematics*, ch. 15 (ed. E. F. Beckenbach), Wiley, New York, pp. 416–55

Whiting, P. D. and Hillier, J. A. (1960). A method for finding the shortest route through a road network. *Operational Res. Quart.*, **11,** 37–40

FEUE WHEP

5604

2 Trees

2.1 TREES

An *acyclic* graph is one that contains no cycles. A *tree* is a connected acyclic graph. The trees on six vertices are shown in figure 2.1.

Theorem 2.1 In a tree, any two vertices are connected by a unique path.

Proof By contradiction. Let G be a tree, and assume that there are two distinct (u, v)-paths P_1 and P_2 in G. Since $P_1 \neq P_2$, there is an edge $e = xy$ of P_1 that is not an edge of P_2. Clearly the graph $(P_1 \cup P_2) - e$ is connected. It therefore contains an (x, y)-path P. But then $P + e$ is a cycle in the acyclic graph G, a contradiction □

The converse of this theorem holds for graphs without loops (exercise 2.1.1).

Observe that all the trees on six vertices (figure 2.1) have five edges. In general we have:

Theorem 2.2 If G is a tree, then $\varepsilon = \nu - 1$.

Proof By induction on ν. When $\nu = 1$, $G \cong K_1$ and $\varepsilon = 0 = \nu - 1$.

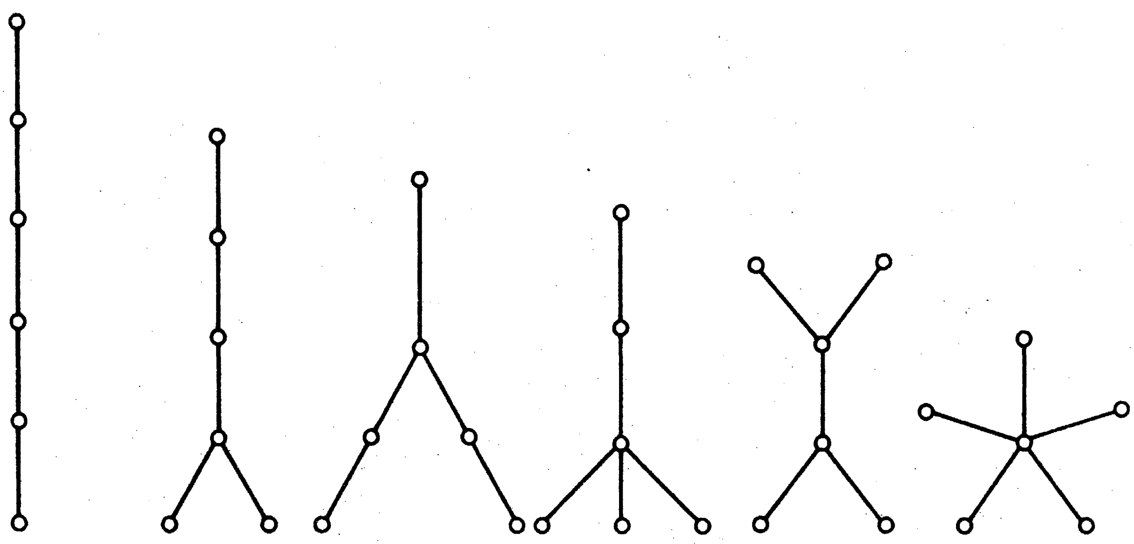


Figure 2.1. The trees on six vertices

FEUE WHEP

5605

Suppose the theorem true for all trees on fewer than ν vertices, and let G be a tree on $\nu \geq 2$ vertices. Let $uv \in E$. Then $G-uv$ contains no (u, v)-path, since uv is the unique (u, v)-path in G. Thus $G-uv$ is disconnected and so (exercise 1.6.8a) $\omega(G-uv)=2$. The components G_1 and G_2 of $G-uv$, being acyclic, are trees. Moreover, each has fewer than ν vertices. Therefore, by the induction hypothesis

$$\varepsilon(G_i)=\nu(G_i)-1 \quad \text{for} \quad i=1,2$$

Thus

$$\varepsilon(G)=\varepsilon(G_1)+\varepsilon(G_2)+1=\nu(G_1)+\nu(G_2)-1=\nu(G)-1 \quad \square$$

Corollary 2.2 Every nontrivial tree has at least two vertices of degree one.

Proof Let G be a nontrivial tree. Then

$$d(v)\geq 1 \quad \text{for all} \quad v \in V$$

Also, by theorems 1.1 and 2.2, we have

$$\sum_{v\in V} d(v)=2\varepsilon=2\nu-2$$

It now follows that $d(v)=1$ for at least two vertices v $\square$

Another, perhaps more illuminating, way of proving corollary 2.2 is to show that the origin and terminus of a longest path in a nontrivial tree both have degree one (see exercise 2.1.2).

Exercises

2.1.1 Show that if any two vertices of a loopless graph G are connected by a unique path, then G is a tree.

2.1.2 Prove corollary 2.2 by showing that the origin and terminus of a longest path in a nontrivial tree both have degree one.

2.1.3 Prove corollary 2.2 by using exercise 1.7.2.

2.1.4 Show that every tree with exactly two vertices of degree one is a path.

2.1.5 Let G be a graph with $\nu-1$ edges. Show that the following three statements are equivalent:

(*a*) G is connected;
(*b*) G is acyclic;
(*c*) G is a tree.

2.1.6 Show that if G is a tree with $\Delta \geq k$, then G has at least k vertices of degree one.

2.1.7 An acyclic graph is also called a *forest*. Show that

(*a*) each component of a forest is a tree;
(*b*) G is a forest if and only if $\varepsilon=\nu-\omega$.

FEUE WHEP

5606

2.1.8 A *centre* of G is a vertex u such that $\max_{v \in V} d(u, v)$ is as small as possible. Show that a tree has either exactly one centre or two, adjacent, centres.

2.1.9 Show that if G is a forest with exactly $2k$ vertices of odd degree, then there are k edge-disjoint paths $P_1, P_2, \ldots, P_k$ in G such that $E(G) = E(P_1) \cup E(P_2) \cup \ldots \cup E(P_k)$.

2.1.10* Show that a sequence $(d_1, d_2, \ldots, d_\nu)$ of positive integers is a degree sequence of a tree if and only if $\sum_{i=1}^{\nu} d_i = 2(\nu - 1)$.

2.1.11 Let T be an arbitrary tree on $k+1$ vertices. Show that if G is simple and $\delta \geq k$ then G has a subgraph isomorphic to T.

2.1.12 A saturated hydrocarbon is a molecule C_mH_n in which every carbon atom has four bonds, every hydrogen atom has one bond, and no sequence of bonds forms a cycle. Show that, for every positive integer m, C_mH_n can exist only if $n = 2m + 2$.

2.2 CUT EDGES AND BONDS

A *cut edge* of G is an edge e such that $\omega(G-e) > \omega(G)$. The graph of figure 2.2 has the three cut edges indicated.

Theorem 2.3 An edge e of G is a cut edge of G if and only if e is contained in no cycle of G.

Proof Let e be a cut edge of G. Since $\omega(G-e) > \omega(G)$, there exist vertices u and v of G that are connected in G but not in $G-e$. There is therefore some (u, v)-path P in G which, necessarily, traverses e. Suppose that x and y are the ends of e, and that x precedes y on P. In $G-e$, u is connected to x by a section of P and y is connected to v by a section of P. If e were in a cycle C, x and y would be connected in $G-e$ by the path $C-e$. Thus, u and v would be connected in $G-e$, a contradiction.

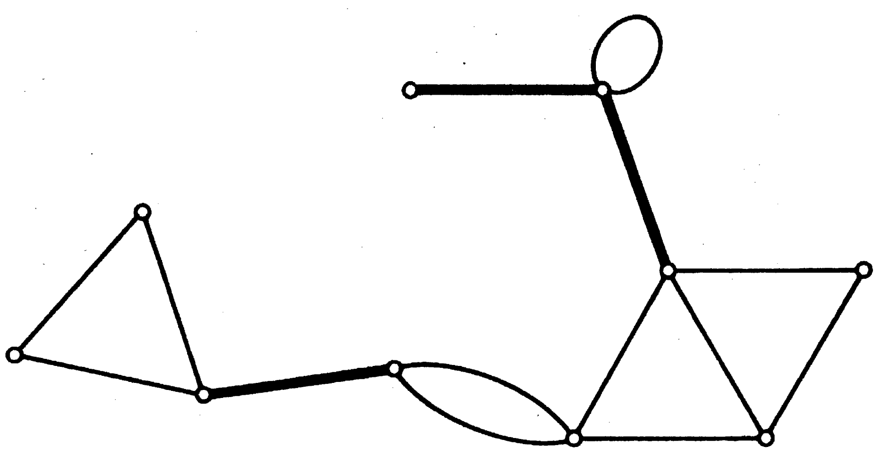


Figure 2.2. The cut edges of a graph

FEUE WHEP

5607

Conversely, suppose that $e = xy$ is not a cut edge of G; thus, $\omega(G-e) = \omega(G)$. Since there is an (x, y)-path (namely xy) in G, x and y are in the same component of G. It follows that x and y are in the same component of $G-e$, and hence that there is an (x, y)-path P in $G-e$. But then e is in the cycle $P+e$ of G □

Theorem 2.4 A connected graph is a tree if and only if every edge is a cut edge.

Proof Let G be a tree and let e be an edge of G. Since G is acyclic, e is contained in no cycle of G and is therefore, by theorem 2.3, a cut edge of G.

Conversely, suppose that G is connected but is not a tree. Then G contains a cycle C. By theorem 2.3, no edge of C can be a cut edge of G □

A *spanning tree* of G is a spanning subgraph of G that is a tree.

Corollary 2.4.1 Every connected graph contains a spanning tree.

Proof Let G be connected and let T be a minimal connected spanning subgraph of G. By definition $\omega(T) = 1$ and $\omega(T-e) > 1$ for each edge e of T. It follows that each edge of T is a cut edge and therefore, by theorem 2.4, that T, being connected, is a tree □

Figure 2.3 depicts a connected graph and one of its spanning trees.

Corollary 2.4.2 If G is connected, then $\varepsilon \geq \nu - 1$.

Proof Let G be connected. By corollary 2.4.1, G contains a spanning tree T. Therefore

$$\varepsilon(G) \geq \varepsilon(T) = \nu(T) - 1 = \nu(G) - 1 \quad \square$$

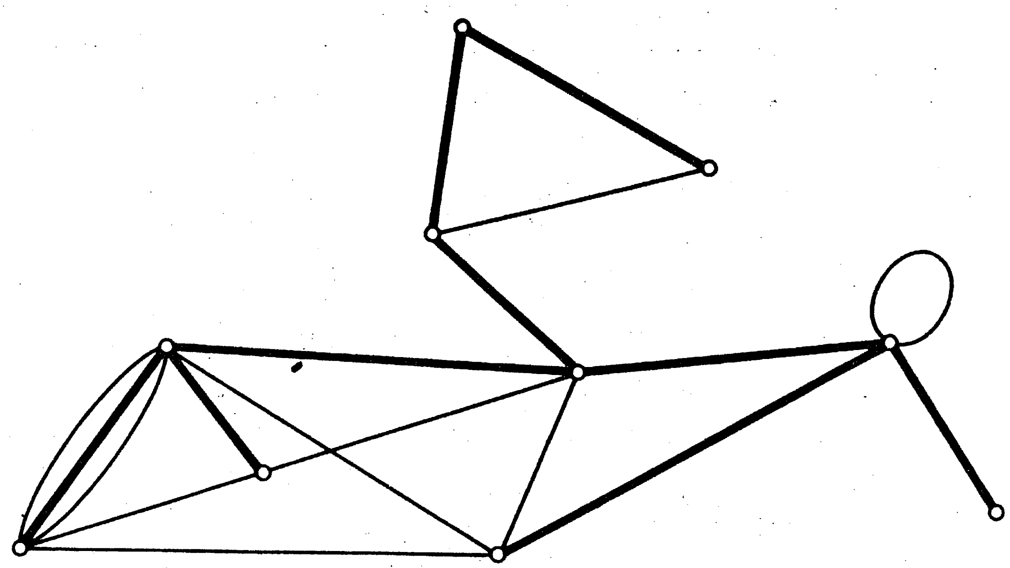
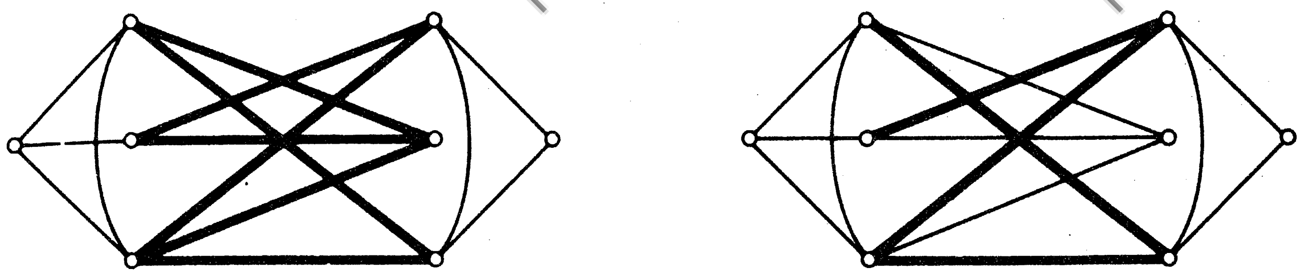


Figure 2.3. A spanning tree in a connected graph

FEUE WHEP

5608



(a) (b)

Figure 2.4. (*a*) An edge cut; (*b*) a bond

Theorem 2.5 Let T be a spanning tree of a connected graph G and let e be an edge of G not in T. Then $T+e$ contains a unique cycle.

Proof Since T is acyclic, each cycle of $T+e$ contains e. Moreover, C is a cycle of $T+e$ if and only if $C-e$ is a path in T connecting the ends of e. By theorem 2.1, T has a unique such path; therefore $T+e$ contains a unique cycle □

For subsets S and S' of V, we denote by $[S, S']$ the set of edges with one end in S and the other in S'. An *edge cut* of G is a subset of E of the form $[S, \bar{S}]$, where S is a nonempty proper subset of V and $\bar{S} = V\backslash S$. A minimal nonempty edge cut of G is called a *bond*; each cut edge e, for instance, gives rise to a bond $\{e\}$. If G is connected, then a bond B of G is a minimal subset of E such that $G-B$ is disconnected. Figure 2.4 indicates an edge cut and a bond in a graph.

If H is a subgraph of G, the *complement of H in G*, denoted by $\bar{H}(G)$, is the subgraph $G-E(H)$. If G is connected, a subgraph of the form $\bar{T}$, where T is a spanning tree, is called a *cotree* of G.

Theorem 2.6 Let T be a spanning tree of a connected graph G, and let e be any edge of T. Then

(i) the cotree $\bar{T}$ contains no bond of G;
(ii) $\bar{T}+e$ contains a unique bond of G.

Proof (i) Let B be a bond of G. Then $G-B$ is disconnected, and so cannot contain the spanning tree T. Therefore B is not contained in $\bar{T}$. (ii) Denote by S the vertex set of one of the two components of $T-e$. The edge cut $B=[S, \bar{S}]$ is clearly a bond of G, and is contained in $\bar{T}+e$. Now, for any $b \in B$, $T-e+b$ is a spanning tree of G. Therefore every bond of G contained in $\bar{T}+e$ must include every such element b. It follows that B is the only bond of G contained in $\bar{T}+e$ □

The relationship between bonds and cotrees is analogous to that between cycles and spanning trees. Statement (i) of theorem 2.6 is the analogue for

5609

bonds of the simple fact that a spanning tree is acyclic, and (ii) is the analogue of theorem 2.5. This 'duality' between cycles and bonds will be further explored in chapter 12 (see also exercise 2.2.10).

Exercises

2.2.1 Show that G is a forest if and only if every edge of G is a cut edge.

2.2.2 Let G be connected and let $e \in E$. Show that

(*a*) e is in every spanning tree of G if and only if e is a cut edge of G;

(*b*) e is in no spanning tree of G if and only if e is a loop of G.

2.2.3 Show that if G is loopless and has exactly one spanning tree T, then $G = T$.

2.2.4 Let F be a maximal forest of G. Show that

(*a*) for every component H of G, $F \cap H$ is a spanning tree of H;

(*b*) $\varepsilon(F) = \nu(G) - \omega(G)$.

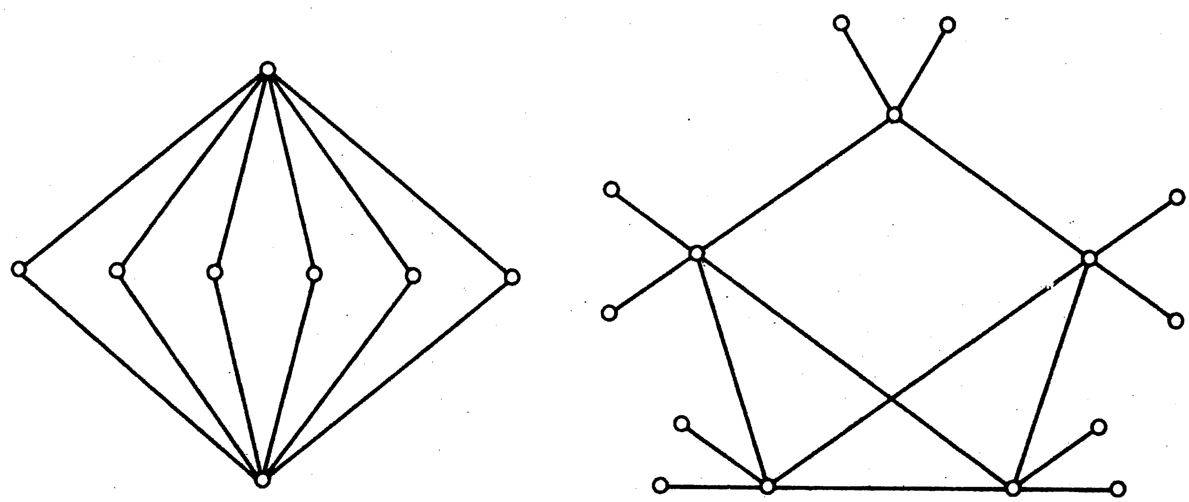
2.2.5 Show that G contains at least $\varepsilon - \nu + \omega$ distinct cycles.

2.2.6 Show that

(*a*) if each degree in G is even, then G has no cut edge;

(*b*) if G is a k-regular bipartite graph with $k \geq 2$, then G has no cut edge.

2.2.7 Find the number of nonisomorphic spanning trees in the following graphs:



2.2.8 Let G be connected and let S be a nonempty proper subset of V. Show that the edge cut $[S, \bar{S}]$ is a bond of G if and only if both $G[S]$ and $G[\bar{S}]$ are connected.

2.2.9 Show that every edge cut is a disjoint union of bonds.

2.2.10 Let B_1 and B_2 be bonds and let C_1 and C_2 be cycles (regarded as

FEUE WHEP

5610

sets of edges) in a graph. Show that

(*a*) $B_1 \Delta B_2$ is a disjoint union of bonds;
(*b*) $C_1 \Delta C_2$ is a disjoint union of cycles,

where Δ denotes symmetric difference;

(*c*) for any edge e, $(B_1 \cup B_2)\backslash\{e\}$ contains a bond;
(*d*) for any edge e, $(C_1 \cup C_2)\backslash\{e\}$ contains a cycle.

2.2.11 Show that if a graph G contains k edge-disjoint spanning trees then, for each partition $(V_1, V_2, \ldots, V_n)$ of V, the number of edges which have ends in different parts of the partition is at least $k(n-1)$.

(Tutte, 1961 and Nash-Williams, 1961 have shown that this necessary condition for G to contain k edge-disjoint spanning trees is also sufficient.)

2.2.12* Let S be an n-element set, and let $\mathscr{A} = \{A_1, A_2, \ldots, A_n\}$ be a family of n distinct subsets of S. Show that there is an element $x \in S$ such that the sets $A_1 \cup \{x\}$, $A_2 \cup \{x\}, \ldots, A_n \cup \{x\}$ are all distinct.

2.3 CUT VERTICES

A vertex v of G is a *cut vertex* if E can be partitioned into two nonempty subsets E_1 and E_2 such that $G[E_1]$ and $G[E_2]$ have just the vertex v in common. If G is loopless and nontrivial, then v is a cut vertex of G if and only if $\omega(G-v) > \omega(G)$. The graph of figure 2.5 has the five cut vertices indicated.

Theorem 2.7 A vertex v of a tree G is a cut vertex of G if and only if $d(v) > 1$.

Proof If $d(v) = 0$, $G \cong K_1$ and, clearly, v is not a cut vertex.

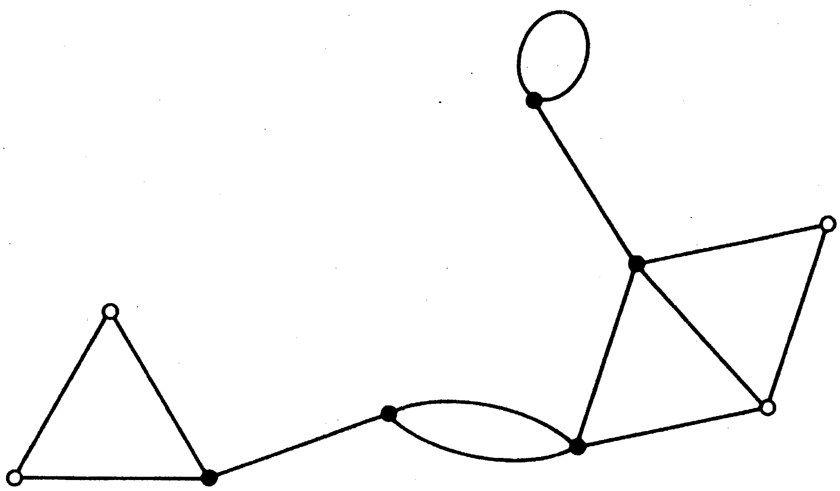


Figure 2.5. The cut vertices of a graph

FEUE WHEP

5611

If $d(v)=1$, $G-v$ is an acyclic graph with $\nu(G-v)-1$ edges, and thus (exercise 2.1.5) a tree. Hence $\omega(G-v)=1=\omega(G)$, and v is not a cut vertex of G.

If $d(v)>1$, there are distinct vertices u and w adjacent to v. The path uvw is a (u, w)-path in G. By theorem 2.1 uvw is the unique (u, w)-path in G. It follows that there is no (u, w)-path in $G-v$, and therefore that $\omega(G-v)> 1=\omega(G)$. Thus v is a cut vertex of G □

Corollary 2.7 Every nontrivial loopless connected graph has at least two vertices that are not cut vertices.

Proof Let G be a nontrivial loopless connected graph. By corollary 2.4.1, G contains a spanning tree T. By corollary 2.2 and theorem 2.7, T has at least two vertices that are not cut vertices. Let v be any such vertex. Then

$$\omega(T-v)=1$$

Since T is a spanning subgraph of G, $T-v$ is a spanning subgraph of $G-v$ and therefore

$$\omega(G-v)\leq\omega(T-v)$$

It follows that $\omega(G-v)=1$, and hence that v is not a cut vertex of G. Since there are at least two such vertices v, the proof is complete □

Exercises

2.3.1 Let G be connected with $\nu\geq 3$. Show that

(*a*) if G has a cut edge, then G has a vertex v such that $\omega(G-v)> \omega(G)$;

(*b*) the converse of (*a*) is not necessarily true.

2.3.2 Show that a simple connected graph that has exactly two vertices which are not cut vertices is a path.

2.4 CAYLEY'S FORMULA

There is a simple and elegant recursive formula for the number of spanning trees in a graph. It involves the operation of contraction of an edge, which we now introduce. An edge e of G is said to be *contracted* if it is deleted and its ends are identified; the resulting graph is denoted by $G\cdot e$. Figure 2.6 illustrates the effect of contracting an edge.

It is clear that if e is a link of G, then

$$\nu(G\cdot e)=\nu(G)-1 \qquad \varepsilon(G\cdot e)=\varepsilon(G)-1 \quad \text{and} \quad \omega(G\cdot e)=\omega(G)$$

Therefore, if T is a tree, so too is $T\cdot e$.

We denote the number of spanning trees of G by $\tau(G)$.

5612

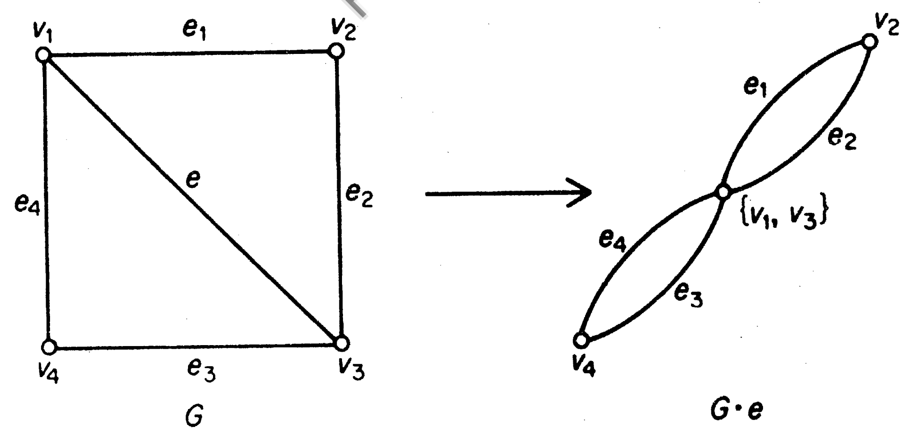


Figure 2.6. Contraction of an edge

Theorem 2.8 If e is a link of G, then $\tau(G) = \tau(G-e) + \tau(G \cdot e)$.

Proof Since every spanning tree of G that does not contain e is also a spanning tree of $G-e$, and conversely, $\tau(G-e)$ is the number of spanning trees of G that do not contain e.

Now to each spanning tree T of G that contains e, there corresponds a spanning tree $T \cdot e$ of $G \cdot e$. This correspondence is clearly a bijection (see figure 2.7). Therefore $\tau(G \cdot e)$ is precisely the number of spanning trees of G that contain e. It follows that $\tau(G) = \tau(G-e) + \tau(G \cdot e)$ □

Figure 2.8 illustrates the recursive calculation of $\tau(G)$ by means of theorem 2.8; the number of spanning trees in a graph is represented symbolically by the graph itself.

Although theorem 2.8 provides a method of calculating the number of spanning trees in a graph, this method is not suitable for large graphs. Fortunately, and rather surprisingly, there is a closed formula for $\tau(G)$ which expresses $\tau(G)$ as a determinant; we shall present this result in chapter 12. In the special case when G is complete, a simple formula for $\tau(G)$ was discovered by Cayley (1889). The proof we give is due to Prüfer (1918).

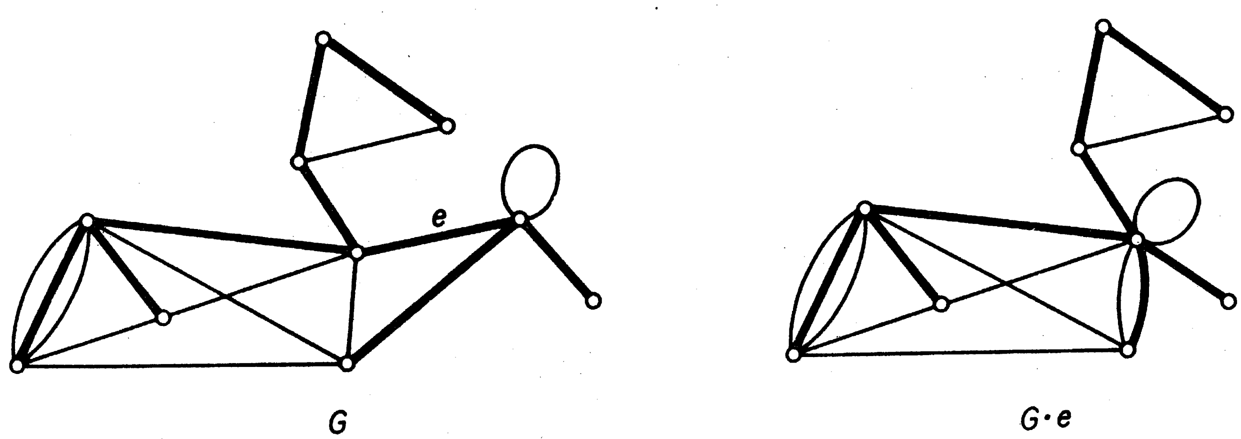


Figure 2.7

5613

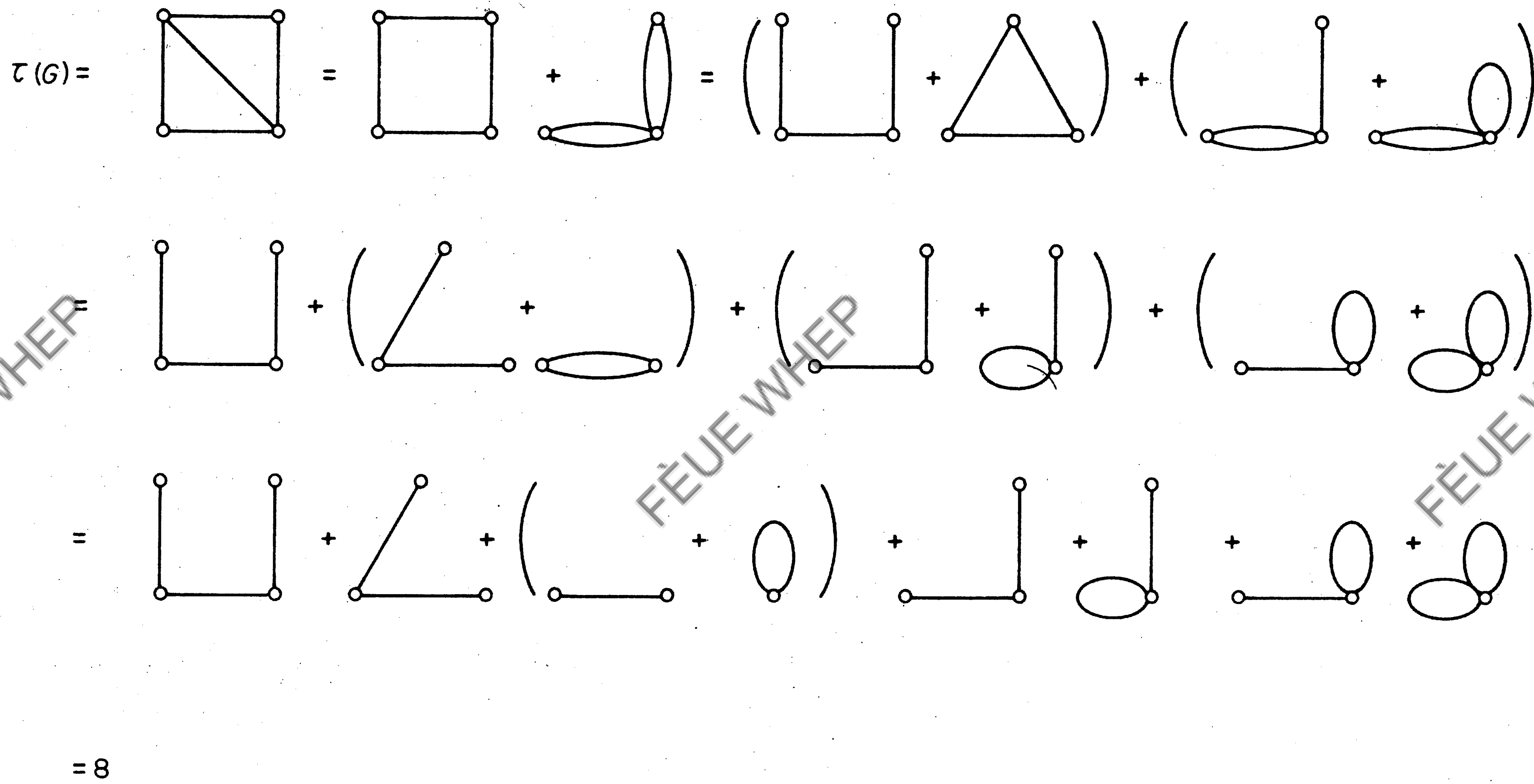


Figure 2.8. Recursive calculation of $\tau(G)$

FEUE WHEP

5614

Theorem 2.9 $\tau(K_n) = n^{n-2}$.

Proof Let the vertex set of K_n be $N = \{1, 2, \ldots, n\}$. We note that n^{n-2} is the number of sequences of length $n-2$ that can be formed from N. Thus, to prove the theorem, it suffices to establish a one–one correspondence between the set of spanning trees of K_n and the set of such sequences.

With each spanning tree T of K_n, we associate a unique sequence $(t_1, t_2, \ldots, t_{n-2})$ as follows. Regarding N as an ordered set, let s_1 be the first vertex of degree one in T; the vertex adjacent to s_1 is taken as t_1. We now delete s_1 from T, denote by s_2 the first vertex of degree one in $T - s_1$, and take the vertex adjacent to s_2 as t_2. This operation is repeated until t_{n-2} has been defined and a tree with just two vertices remains; the tree in figure 2.9, for instance, gives rise to the sequence (4, 3, 5, 3, 4, 5). It can be seen that different spanning trees of K_n determine difference sequences.

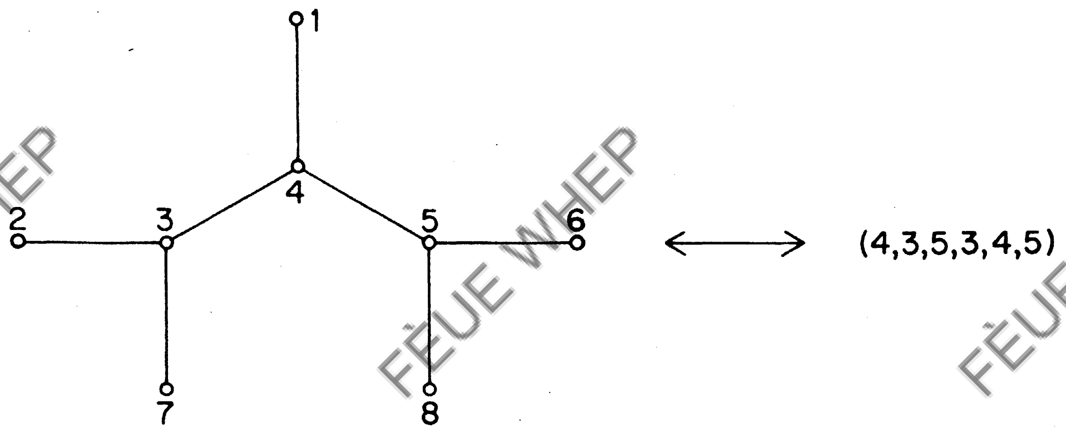


Figure 2.9

The reverse procedure is equally straightforward. Observe, first, that any vertex v of T occurs $d_T(v) - 1$ times in $(t_1, t_2, \ldots, t_{n-2})$. Thus the vertices of degree one in T are precisely those that do not appear in this sequence. To reconstruct T from $(t_1, t_2, \ldots, t_{n-2})$, we therefore proceed as follows. Let s_1 be the first vertex of N not in $(t_1, t_2, \ldots, t_{n-2})$; join s_1 to t_1. Next, let s_2 be the first vertex of $N \backslash \{s_1\}$ not in $(t_2, \ldots, t_{n-2})$, and join s_2 to t_2. Continue in this way until the $n-2$ edges $s_1t_1, s_2t_2, \ldots, s_{n-2}t_{n-2}$ have been determined. T is now obtained by adding the edge joining the two remaining vertices of $N \backslash \{s_1, s_2, \ldots, s_{n-2}\}$. It is easily verified that different sequences give rise to different spanning trees of K_n. We have thus established the desired one–one correspondence □

Note that n^{n-2} is not the number of nonisomorphic spanning trees of K_n, but the number of distinct spanning trees of K_n; there are just six nonisomorphic spanning trees of K_6 (see figure 2.1), whereas there are $6^4 = 1296$ distinct spanning trees of K_6.

FEUE WHEP

5615

Exercises

2.4.1 Using the recursion formula of theorem 2.8, evaluate the number of spanning trees in $K_{3,3}$.

2.4.2* A *wheel* is a graph obtained from a cycle by adding a new vertex and edges joining it to all the vertices of the cycle; the new edges are called the *spokes* of the wheel. Obtain an expression for the number of spanning trees in a wheel with n spokes.

2.4.3 Draw all sixteen spanning trees of K_4.

2.4.4 Show that if e is an edge of K_n, then $\tau(K_n - e) = (n-2)n^{n-3}$.

2.4.5 (*a*) Let H be a graph in which every two adjacent vertices are joined by k edges and let G be the underlying simple graph of H. Show that $\tau(H) = k^{\nu-1}\tau(G)$.

(*b*) Let H be the graph obtained from a graph G when each edge of G is replaced by a path of length k. Show that $\tau(H) = k^{\epsilon-\nu+1}\tau(G)$.

(*c*) Deduce from (*b*) that $\tau(K_{2,n}) = n2^{n-1}$.

APPLICATIONS

2.5 THE CONNECTOR PROBLEM

A railway network connecting a number of towns is to be set up. Given the cost c_{ij} of constructing a direct link between towns v_i and v_j, design such a network to minimise the total cost of construction. This is known as the *connector problem*.

By regarding each town as a vertex in a weighted graph with weights $w(v_iv_j) = c_{ij}$, it is clear that this problem is just that of finding, in a weighted graph G, a connected spanning subgraph of minimum weight. Moreover, since the weights represent costs, they are certainly non-negative, and we may therefore assume that such a minimum-weight spanning subgraph is a spanning tree T of G. A minimum-weight spanning tree of a weighted graph will be called an *optimal tree*; the spanning tree indicated in the weighted graph of figure 2.10 is an optimal tree (exercise 2.5.1).

We shall now present a good algorithm for finding an optimal tree in a nontrivial weighted connected graph, thereby solving the connector problem.

Consider, first, the case when each weight $w(e) = 1$. An optimal tree is then a spanning tree with as few edges as possible. Since each spanning tree of a graph has the same number of edges (theorem 2.2), in this special case we merely need to construct some spanning tree of the graph. A simple

5616

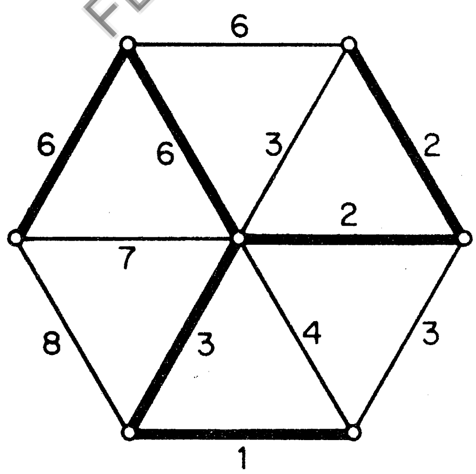


Figure 2.10. An optimal tree in a weighted graph

inductive algorithm for finding such a tree is the following:

1. Choose a link e_1.
2. If edges $e_1, e_2, \ldots, e_i$ have been chosen, then choose e_{i+1} from $E \backslash \{e_1, e_2, \ldots, e_i\}$ in such a way that $G[\{e_1, e_2, \ldots, e_{i+1}\}]$ is acyclic.
3. Stop when step 2 cannot be implemented further.

This algorithm works because a maximal acyclic subgraph of a connected graph is necessarily a spanning tree. It was extended by Kruskal (1956) to solve the general problem; his algorithm is valid for arbitrary real weights.

Kruskal's Algorithm

1. Choose a link e_1 such that $w(e_1)$ is as small as possible.
2. If edges $e_1, e_2, \ldots, e_i$ have been chosen, then choose an edge e_{i+1} from $E \backslash \{e_1, e_2, \ldots, e_i\}$ in such a way that
   (i) $G[\{e_1, e_2, \ldots, e_{i+1}\}]$ is acyclic;
   (ii) $w(e_{i+1})$ is as small as possible subject to (i).
3. Stop when step 2 cannot be implemented further.

As an example, consider the table of airline distances in miles between six of the largest cities in the world, London, Mexico City, New York, Paris, Peking and Tokyo:

	L	MC	NY	Pa	Pe	T
L	—	5558	3469	214	5074	5959
MC	5558	—	2090	5725	7753	7035
NY	3469	2090	—	3636	6844	6757
Pa	214	5725	3636	—	5120	6053
Pe	5074	7753	6844	5120	—	1307
T	5959	7035	6757	6053	1307	—

5617

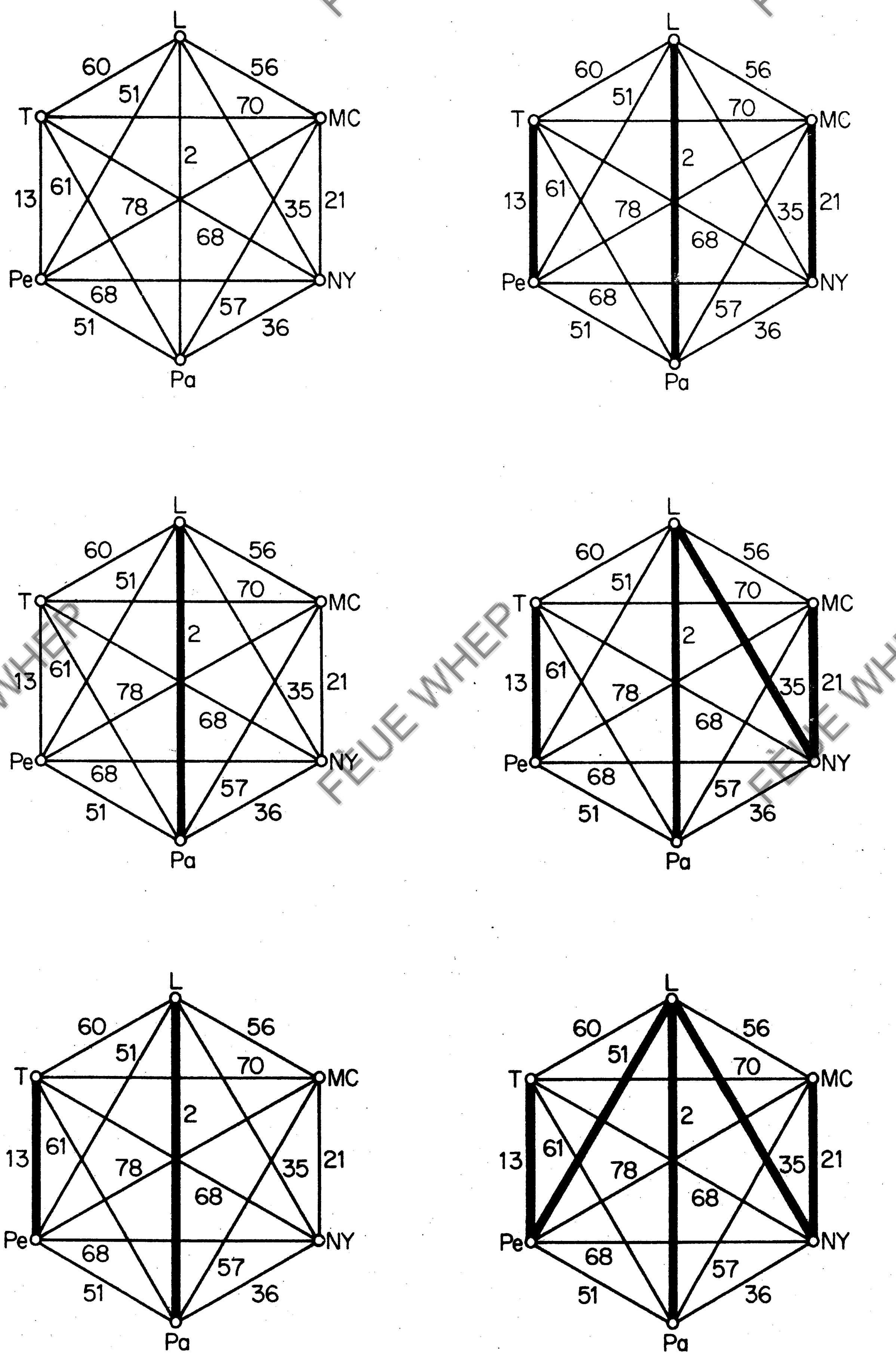


Figure 2.11

FEUE WHEP

5618

This table determines a weighted complete graph with vertices L, MC, NY, Pa, Pe and T. The construction of an optimal tree in this graph is shown in figure 2.11 (where, for convenience, distances are given in hundreds of miles).

Kruskal's algorithm clearly produces a spanning tree (for the same reason that the simpler algorithm above does). The following theorem ensures that such a tree will always be optimal.

Theorem 2.10 Any spanning tree $T^* = G[\{e_1, e_2, \ldots, e_{\nu-1}\}]$ constructed by Kruskal's algorithm is an optimal tree.

Proof By contradiction. For any spanning tree T of G other than T^*, denote by $f(T)$ the smallest value of i such that e_i is not in T. Now assume that T^* is not an optimal tree, and let T be an optimal tree such that $f(T)$ is as large as possible.

Suppose that $f(T) = k$; this means that $e_1, e_2, \ldots, e_{k-1}$ are in both T and T^*, but that e_k is not in T. By theorem 2.5, $T + e_k$ contains a unique cycle C. Let e'_k be an edge of C that is in T but not in T^*. By theorem 2.3, e'_k is not a cut edge of $T + e_k$. Hence $T' = (T + e_k) - e'_k$ is a connected graph with $\nu - 1$ edges, and therefore (exercise 2.1.5) is another spanning tree of G. Clearly

$$w(T') = w(T) + w(e_k) - w(e'_k) \tag{2.1}$$

Now, in Kruskal's algorithm, e_k was chosen as an edge with the smallest weight such that $G[\{e_1, e_2, \ldots, e_k\}]$ was acyclic. Since $G[\{e_1, e_2, \ldots, e_{k-1}, e'_k\}]$ is a subgraph of T, it is also acyclic. We conclude that

$$w(e'_k) \geq w(e_k) \tag{2.2}$$

Combining (2.1) and (2.2) we have

$$w(T') \leq w(T)$$

and so T', too, is an optimal tree. However

$$f(T') > k = f(T)$$

contradicting the choice of T. Therefore $T = T^*$, and T^* is indeed an optimal tree □

A flow diagram for Kruskal's algorithm is shown in figure 2.12. The edges are first sorted in order of increasing weight (box 1); this takes about $\varepsilon \log \varepsilon$ computations (see Knuth, 1973). Box 2 just checks to see how many edges have been chosen. (S is the set of edges already chosen and i is their number.) When $i = \nu - 1$, $S = \{e_1, e_2, \ldots, e_{\nu-1}\}$ is the edge set of an optimal tree T^* of G. In box 3, to check if $G[S \cup \{a_j\}]$ is acyclic, one must ascertain whether the ends of a_j are in different components of the forest $G[S]$ or not. This can be achieved in the following way. The vertices are labelled so that, at any stage, two vertices belong to the same component of $G[S]$ if and only

5619

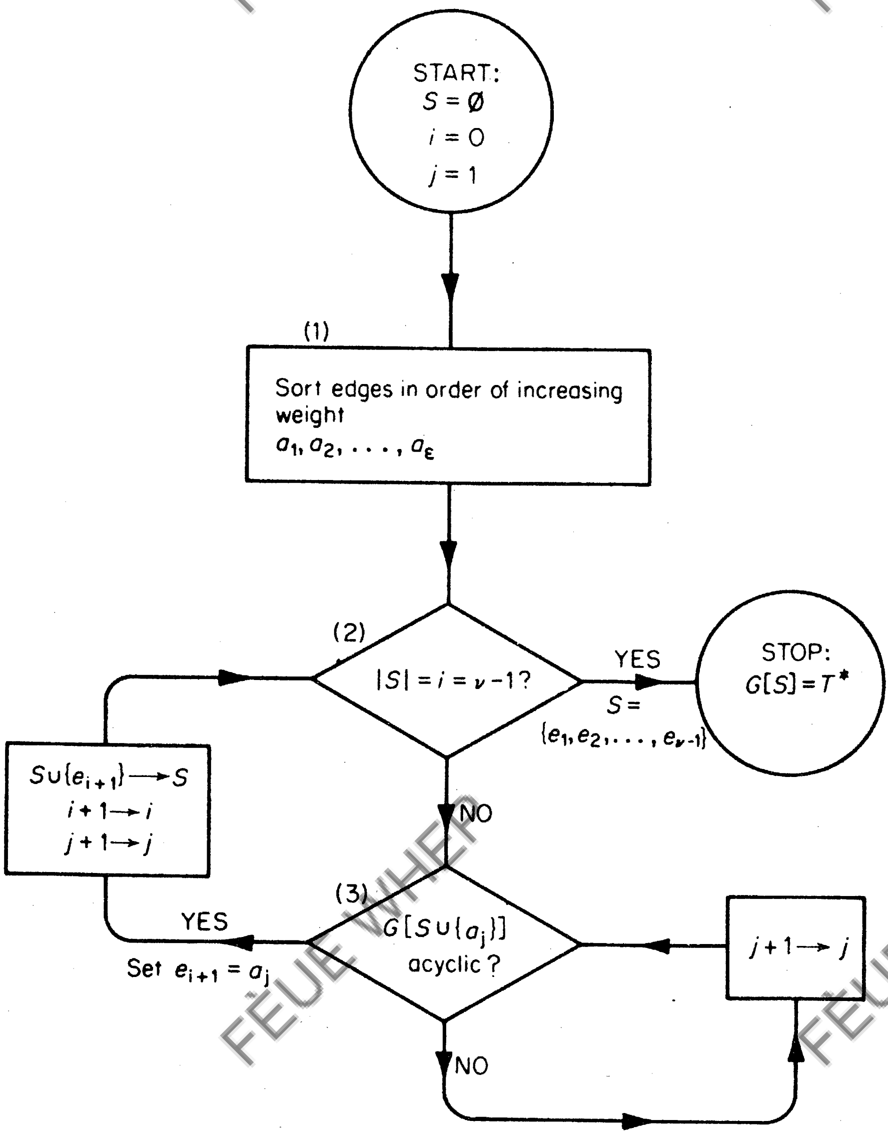


Figure 2.12. Kruskal's algorithm

if they have the same label; initially, vertex v_l is assigned the label l, $1 \leq l \leq \nu$. With this labelling scheme, $G[S \cup \{a_j\}]$ is acyclic if and only if the ends of a_j have different labels. If this is the case, a_j is taken as e_{i+1}; otherwise, a_j is discarded and a_{j+1}, the next candidate for e_{i+1}, is tested. Once e_{i+1} has been added to S, the vertices in the two components of $G[S]$ that contain the ends of e_{i+1} are relabelled with the smaller of their two labels. For each edge, one comparison suffices to check whether its ends have the same or different labels; this takes ε computations. After edge e_{i+1} has been added to S, the relabelling of vertices takes at most ν comparisons; hence, for all $\nu - 1$ edges $e_1, e_2, \ldots, e_{\nu-1}$ we need $\nu(\nu-1)$ computations. Kruskal's algorithm is therefore a good algorithm.

Exercises

2.5.1 Show, by applying Kruskal's algorithm, that the tree indicated in figure 2.10 is indeed optimal.

5620

2.5.2 Adapt Kruskal's algorithm to solve the *connector problem with preassignments*: construct, at minimum cost, a network linking a number of towns, with the additional requirement that certain selected pairs of towns be directly linked.

2.5.3 Can Kruskal's algorithm be adapted to find

(*a*) a *maximum*-weight tree in a weighted connected graph?
(*b*) a minimum-weight maximal *forest* in a weighted graph?

If so, how?

2.5.4 Show that the following Kruskal-type algorithm does not necessarily yield a minimum-weight spanning *path* in a weighted complete graph:

1. Choose a link e_1 such that $w(e_1)$ is as small as possible.
2. If edges $e_1, e_2, \ldots, e_i$ have been chosen, then choose an edge e_{i+1} from $E\backslash\{e_1, e_2, \ldots, e_i\}$ in such a way that
   (i) $G[\{e_1, e_2, \ldots, e_{i+1}\}]$ is a union of disjoint paths;
   (ii) $w(e_{i+1})$ is as small as possible subject to (i).
3. Stop when step 2 cannot be implemented further.

2.5.5 The *tree graph* of a connected graph G is the graph whose vertices are the spanning trees $T_1, T_2, \ldots, T_\tau$ of G, with T_i and T_j joined if and only if they have exactly $\nu-2$ edges in common. Show that the tree graph of any connected graph is connected.

FEUE WHEP

REFERENCES

Cayley, A. (1889). A theorem on trees. *Quart. J. Math.*, **23,** 376–78

Knuth, D. E. (1973). *The Art of Computer Programming*, vol. 3: Sorting and Searching, Addison-Wesley, Reading, Mass., p. 184

Kruskal, J. B. Jr. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, **7,** 48–50

Nash-Williams, C. St. J. A. (1961). Edge-disjoint spanning trees of finite graphs. *J. London Math. Soc.*, **36,** 445–50

Prüfer, H. (1918). Neuer Beweis eines Satzes über Permutationen. *Arch. Math. Phys.*, **27,** 742–44

Tutte, W. T. (1961). On the problem of decomposing a graph into *n* connected factors. *J. London Math. Soc.*, **36,** 221–30

5621

3 Connectivity

3.1 CONNECTIVITY

In section 1.6 we introduced the concept of connection in graphs. Consider, now, the four connected graphs of figure 3.1.

G_1 is a tree, a minimal connected graph; deleting any edge disconnects it. G_2 cannot be disconnected by the deletion of a single edge, but can be disconnected by the deletion of one vertex, its cut vertex. There are no cut edges or cut vertices in G_3, but even so G_3 is clearly not as well connected as G_4, the complete graph on five vertices. Thus, intuitively, each successive graph is more strongly connected than the previous one. We shall now define two parameters of a graph, its connectivity and edge connectivity, which measure the extent to which it is connected.

A *vertex cut* of G is a subset V' of V such that $G-V'$ is disconnected. A k*-vertex cut* is a vertex cut of k elements. A complete graph has no vertex cut; in fact, the only graphs which do not have vertex cuts are those that contain complete graphs as spanning subgraphs. If G has at least one pair of distinct nonadjacent vertices, the *connectivity* $\kappa(G)$ of G is the minimum k for which G has a k-vertex cut; otherwise, we define $\kappa(G)$ to be $\nu-1$. Thus $\kappa(G)=0$ if G is either trivial or disconnected. G is said to be k*-connected* if $\kappa(G)\geq k$. All nontrivial connected graphs are 1-connected.

Recall that an edge cut of G is a subset of E of the form $[S, \bar{S}]$, where S is a nonempty proper subset of V. A k*-edge cut* is an edge cut of k elements. If G is nontrivial and E' is an edge cut of G, then $G-E'$ is disconnected; we then define the *edge connectivity* $\kappa'(G)$ of G to be the minimum k for which G has a k-edge cut. If G is trivial, $\kappa'(G)$ is defined to be zero. Thus $\kappa'(G)=0$ if G is either trivial or disconnected, and $\kappa'(G)=1$ if G is a connected graph with a cut edge. G is said to be k*-edge-connected* if $\kappa'(G)\geq k$. All nontrivial connected graphs are 1-edge-connected.

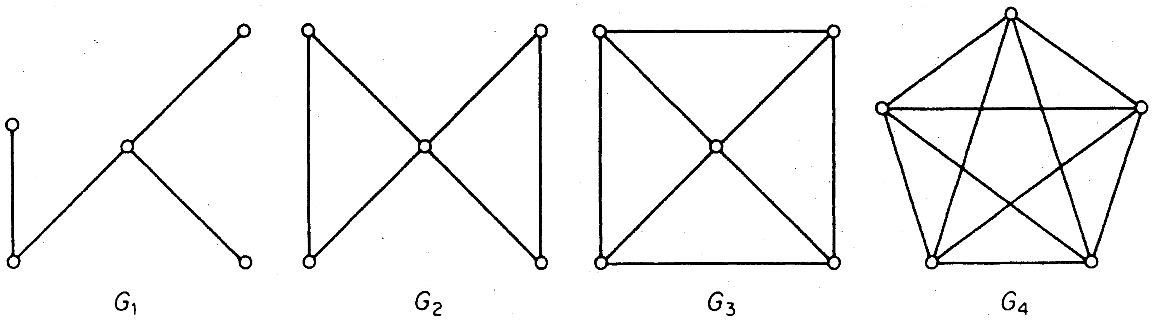


Figure 3.1

FEUE WHEP

5622

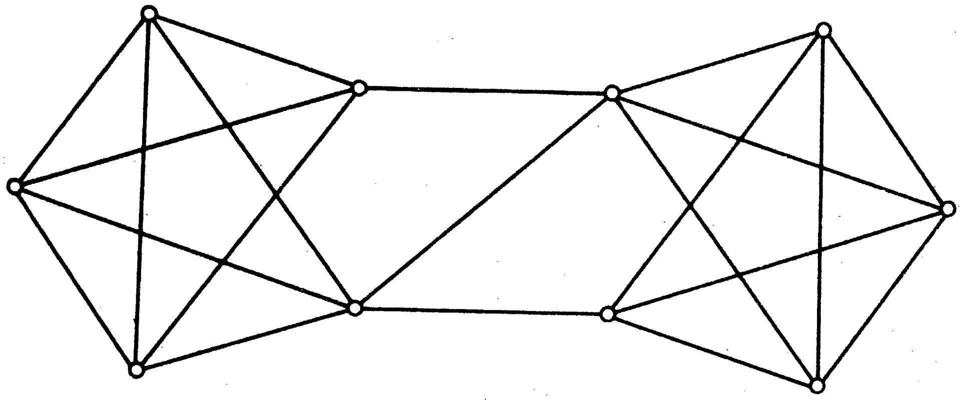


Figure 3.2

Theorem 3.1 $\kappa \leq \kappa' \leq \delta$.

Proof If G is trivial, then $\kappa' = 0 \leq \delta$. Otherwise, the set of links incident with a vertex of degree δ constitute a δ-edge cut of G. It follows that $\kappa' \leq \delta$.

We prove that $\kappa \leq \kappa'$ by induction on κ'. The result is true if $\kappa' = 0$, since then G must be either trivial or disconnected. Suppose that it holds for all graphs with edge connectivity less than k, let G be a graph with $\kappa'(G) = k > 0$, and let e be an edge in a k-edge cut of G. Setting $H = G - e$, we have $\kappa'(H) = k - 1$ and so, by the induction hypothesis, $\kappa(H) \leq k - 1$.

If H contains a complete graph as a spanning subgraph, then so does G and

$$\kappa(G) = \kappa(H) \leq k - 1$$

Otherwise, let S be a vertex cut of H with $\kappa(H)$ elements. Since $H - S$ is disconnected, either $G - S$ is disconnected, and then

$$\kappa(G) \leq \kappa(H) \leq k - 1$$

or else $G - S$ is connected and e is a cut edge of $G - S$. In this latter case, either $\nu(G - S) = 2$ and

$$\kappa(G) \leq \nu(G) - 1 = \kappa(H) + 1 \leq k$$

or (exercise 2.3.1a) $G - S$ has a 1-vertex cut $\{v\}$, implying that $S \cup \{v\}$ is a vertex cut of G and

$$\kappa(G) \leq \kappa(H) + 1 \leq k$$

Thus in each case we have $\kappa(G) \leq k = \kappa'(G)$. The result follows by the principle of induction □

The inequalities in theorem 3.1 are often strict. For example, the graph G of figure 3.2 has $\kappa = 2$, $\kappa' = 3$ and $\delta = 4$.

5623

Exercises

3.1.1 (*a*) Show that if G is k-edge-connected, with $k > 0$, and if E' is a set of k edges of G, then $\omega(G - E') \leq 2$.
(*b*) For $k > 0$, find a k-connected graph G and a set V' of k vertices of G such that $\omega(G - V') > 2$.

3.1.2 Show that if G is k-edge-connected, then $\varepsilon \geq k\nu/2$.

3.1.3 (*a*) Show that if G is simple and $\delta \geq \nu - 2$, then $\kappa = \delta$.
(*b*) Find a simple graph G with $\delta = \nu - 3$ and $\kappa < \delta$.

3.1.4 (*a*) Show that if G is simple and $\delta \geq \nu/2$, then $\kappa' = \delta$.
(*b*) Find a simple graph G with $\delta = [(\nu/2) - 1]$ and $\kappa' < \delta$.

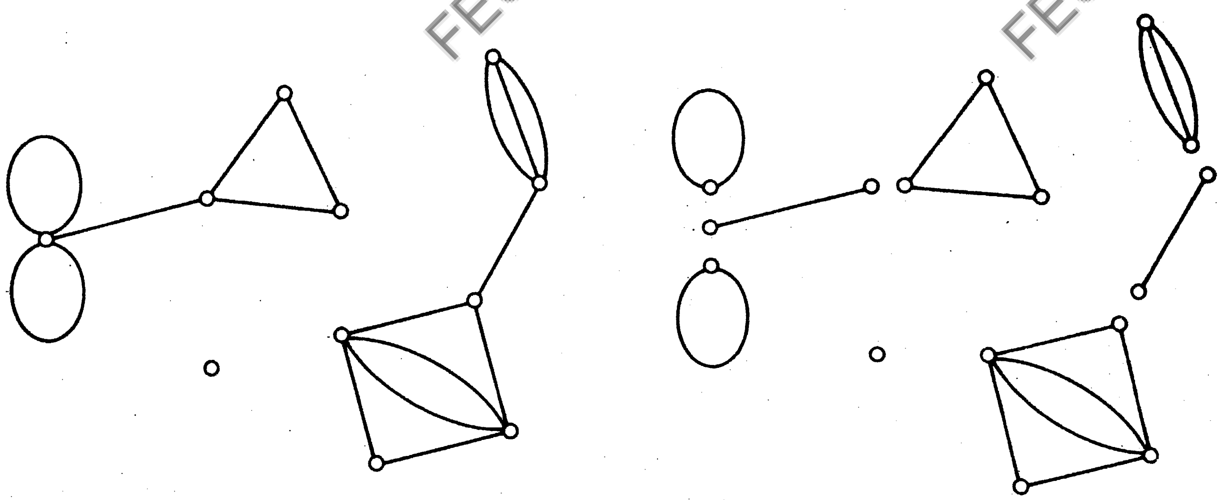
3.1.5 Show that if G is simple and $\delta \geq (\nu + k - 2)/2$, then G is k-connected.

3.1.6 Show that if G is simple and 3-regular, then $\kappa = \kappa'$.

3.1.7 Show that if l, m and n are integers such that $0 < l \leq m \leq n$, then there exists a simple graph G with $\kappa = l$, $\kappa' = m$, and $\delta = n$.
(G. Chartrand and F. Harary)

3.2 BLOCKS

A connected graph that has no cut vertices is called a *block*. Every block with at least three vertices is 2-connected. A *block of a graph* is a subgraph that is a block and is maximal with respect to this property. Every graph is the union of its blocks; this is illustrated in figure 3.3.



(a) (b)

Figure 3.3. (*a*) G; (*b*) the blocks of G

A family of paths in G is said to be *internally-disjoint* if no vertex of G is an internal vertex of more than one path of the family. The following theorem is due to Whitney (1932).

Theorem 3.2 A graph G with $\nu \geq 3$ is 2-connected if and only if any two vertices of G are connected by at least two internally-disjoint paths.

5624

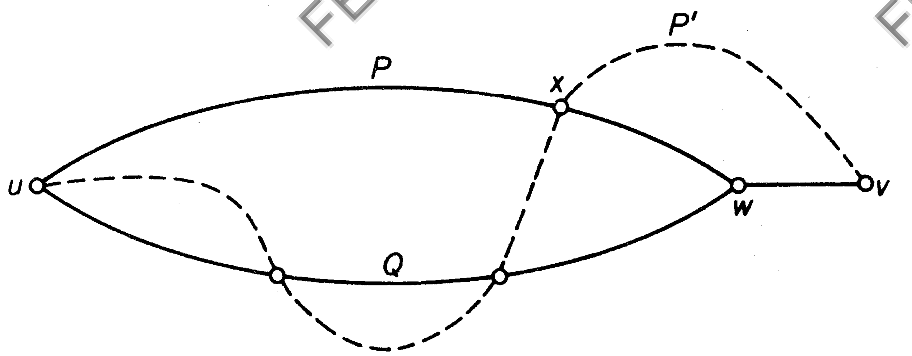


Figure 3.4

Proof If any two vertices of G are connected by at least two internally-disjoint paths then, clearly, G is connected and has no 1-vertex cut. Hence G is 2-connected.

Conversely, let G be a 2-connected graph. We shall prove, by induction on the distance $d(u, v)$ between u and v, that any two vertices u and v are connected by at least two internally-disjoint paths.

Suppose, first, that $d(u, v) = 1$. Then, since G is 2-connected, the edge uv is not a cut edge and therefore, by theorem 2.3, it is contained in a cycle. It follows that u and v are connected by two internally-disjoint paths in G.

Now assume that the theorem holds for any two vertices at distance less than k, and let $d(u, v) = k \geq 2$. Consider a (u, v)-path of length k, and let w be the vertex that precedes v on this path. Since $d(u, w) = k - 1$, it follows from the induction hypothesis that there are two internally-disjoint (u, w)-paths P and Q in G. Also, since G is 2-connected, $G - w$ is connected and so contains a (u, v)-path P'. Let x be the last vertex of P' that is also in $P \cup Q$ (see figure 3.4). Since u is in $P \cup Q$, there is such an x; we do not exclude the possibility that $x = v$.

We may assume, without loss of generality, that x is in P. Then G has two internally-disjoint (u, v)-paths, one composed of the section of P from u to x together with the section of P' from x to v, and the other composed of Q together with the path wv □

Corollary 3.2.1 If G is 2-connected, then any two vertices of G lie on a common cycle.

Proof This follows immediately from theorem 3.2 since two vertices lie on a common cycle if and only if they are connected by two internally-disjoint paths □

It is convenient, now, to introduce the operation of subdivision of an edge. An edge e is said to be *subdivided* when it is deleted and replaced by a path of length two connecting its ends, the internal vertex of this path being a new vertex. This is illustrated in figure 3.5.

5625

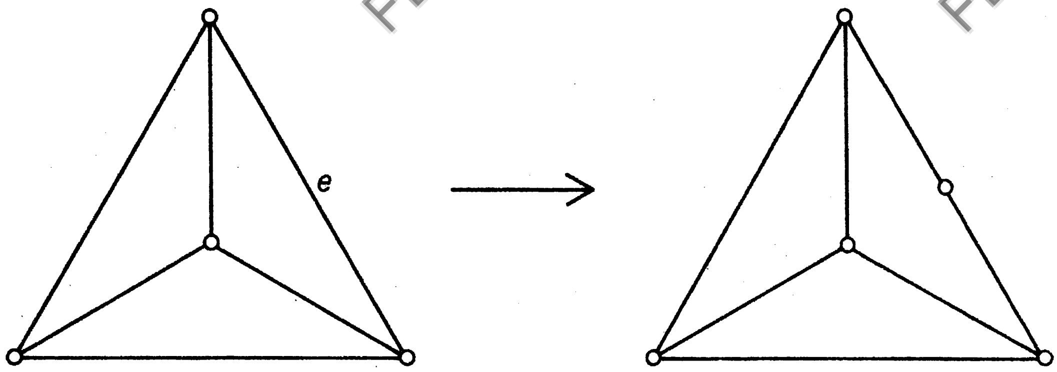


Figure 3.5. Subdivision of an edge

It can be seen that the class of blocks with at least three vertices is closed under the operation of subdivision. The proof of the next corollary uses this fact.

Corollary 3.2.2 If G is a block with $\nu \geq 3$, then any two edges of G lie on a common cycle.

Proof Let G be a block with $\nu \geq 3$, and let e_1 and e_2 be two edges of G. Form a new graph G' by subdividing e_1 and e_2, and denote the new vertices by v_1 and v_2. Clearly, G' is a block with at least five vertices, and hence is 2-connected. It follows from corollary 3.2.1 that v_1 and v_2 lie on a common cycle of G'. Thus e_1 and e_2 lie on a common cycle of G (see figure 3.6) □

Theorem 3.2 has a generalisation to k-connected graphs, known as *Menger's theorem*: a graph G with $\nu \geq k+1$ is k-connected if and only if any two distinct vertices of G are connected by at least k internally-disjoint paths. There is also an edge analogue of this theorem: a graph G is k-edge-connected if and only if any two distinct vertices of G are connected

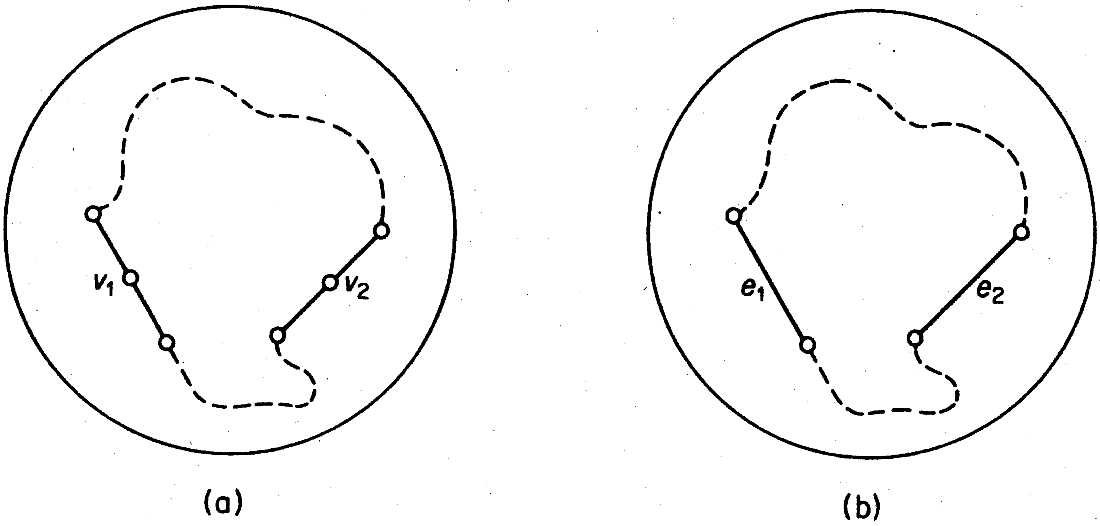


Figure 3.6. (*a*) G'; (*b*) G

FEUE WHEP

5626

by at least k edge-disjoint paths. Proofs of these theorems will be given in chapter 11.

Exercises

3.2.1 Show that a graph is 2-edge-connected if and only if any two vertices are connected by at least two edge-disjoint paths.

3.2.2 Give an example to show that if P is a (u, v)-path in a 2-connected graph G, then G does not necessarily contain a (u, v)-path Q internally-disjoint from P.

3.2.3 Show that if G has no even cycles, then each block of G is either K_1 or K_2, or an odd cycle.

3.2.4 Show that a connected graph which is not a block has at least two blocks that each contain exactly one cut vertex.

3.2.5 Show that the number of blocks in G is equal to $\omega + \sum_{v \in V} (b(v) - 1)$, where $b(v)$ denotes the number of blocks of G containing v.

3.2.6* Let G be a 2-connected graph and let X and Y be disjoint subsets of V, each containing at least two vertices. Show that G contains disjoint paths P and Q such that

(i) the origins of P and Q belong to X,
(ii) the termini of P and Q belong to Y, and
(iii) no internal vertex of P or Q belongs to $X \cup Y$.

3.2.7* A nonempty graph G is *κ-critical* if, for every edge e, $\kappa(G-e) < \kappa(G)$.

(*a*) Show that every κ-critical 2-connected graph has a vertex of degree two.
(Halin, 1969 has shown that, in general, every κ-critical k-connected graph has a vertex of degree k.)

(*b*) Show that if G is a κ-critical 2-connected graph with $\nu \geq 4$, then $\varepsilon \leq 2\nu - 4$. (G. A. Dirac)

3.2.8 Describe a good algorithm for finding the blocks of a graph.

APPLICATIONS

3.3 CONSTRUCTION OF RELIABLE COMMUNICATION NETWORKS

If we think of a graph as representing a communication network, the connectivity (or edge connectivity) becomes the smallest number of communication stations (or communication links) whose breakdown would jeopardise communication in the system. The higher the connectivity and edge connectivity, the more reliable the network. From this point of view, a

FEUE WHEP

5627

tree network, such as the one obtained by Kruskal's algorithm, is not very reliable, and one is led to consider the following generalisation of the connector problem.

Let k be a given positive integer and let G be a weighted graph. Determine a minimum-weight k-connected spanning subgraph of G.

For $k = 1$, this problem reduces to the connector problem, which can be solved by Kruskal's algorithm. For values of k greater than one, the problem is unsolved and is known to be difficult. However, if G is a complete graph in which each edge is assigned unit weight, then the problem has a simple solution which we now present.

Observe that, for a weighted complete graph on n vertices in which each edge is assigned unit weight, a minimum-weight m-connected spanning subgraph is simply an m-connected graph on n vertices with as few edges as possible. We shall denote by $f(m, n)$ the least number of edges that an m-connected graph on n vertices can have. (It is, of course, assumed that $m < n$.) By theorems 3.1 and 1.1

$$f(m, n) \geq \{mn/2\} \tag{3.1}$$

We shall show that equality holds in (3.1) by constructing an m-connected graph $H_{m,n}$ on n vertices that has exactly $\{mn/2\}$ edges. The structure of $H_{m,n}$ depends on the parities of m and n; there are three cases.

Case 1 m even. Let $m = 2r$. Then $H_{2r,n}$ is constructed as follows. It has vertices $0, 1, \ldots, n-1$ and two vertices i and j are joined if $i - r \leq j \leq i + r$ (where addition is taken modulo n). $H_{4,8}$ is shown in figure 3.7a.

Case 2 m odd, n even. Let $m = 2r + 1$. Then $H_{2r+1,n}$ is constructed by first drawing $H_{2r,n}$ and then adding edges joining vertex i to vertex $i + (n/2)$ for $1 \leq i \leq n/2$. $H_{5,8}$ is shown in figure 3.7b.

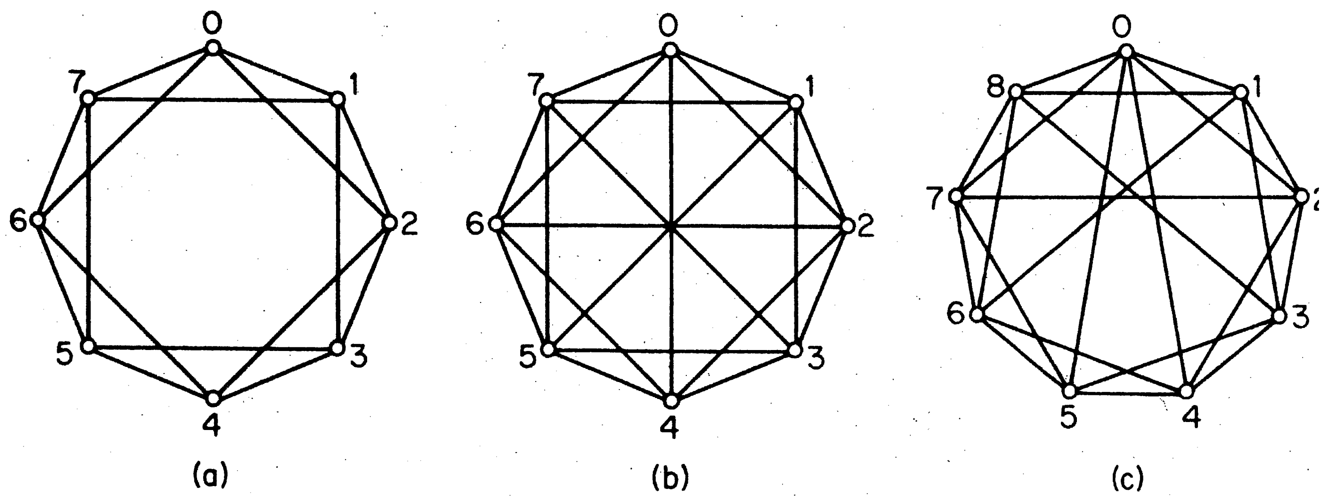


Figure 3.7. (a) $H_{4,8}$; (b) $H_{5,8}$; (c) $H_{5,9}$

FEUE WHEP

5628

Case 3 m odd, n odd. Let $m = 2r+1$. Then $H_{2r+1,n}$ is constructed by first drawing $H_{2r,n}$ and then adding edges joining vertex 0 to vertices $(n-1)/2$ and $(n+1)/2$ and vertex i to vertex $i+(n+1)/2$ for $1 \le i < (n-1)/2$. $H_{5,9}$ is shown in figure 3.7*c*.

Theorem 3.3 (Harary, 1962) The graph $H_{m,n}$ is m-connected.

Proof Consider the case $m = 2r$. We shall show that $H_{2r,n}$ has no vertex cut of fewer than $2r$ vertices. If possible, let V' be a vertex cut with $|V'| < 2r$. Let i and j be vertices belonging to different components of $H_{2r,n} - V'$. Consider the two sets of vertices

$$S = \{i, i+1, \ldots, j-1, j\}$$

and

$$T = \{j, j+1, \ldots, i-1, i\}$$

where addition is taken modulo n. Since $|V'| < 2r$, we may assume, without loss of generality, that $|V' \cap S| < r$. Then there is clearly a sequence of distinct vertices in $S \backslash V'$ which starts with i, ends with j, and is such that the difference between any two consecutive terms is at most r. But such a sequence is an (i, j)-path in $H_{2r,n} - V'$, a contradiction. Hence $H_{2r,n}$ is $2r$-connected.

The case $m = 2r+1$ is left as an exercise (exercise 3.3.1) □

It is easy to see that $\varepsilon(H_{m,n}) = \{mn/2\}$. Thus, by theorem 3.3,

$$f(m, n) \le \{mn/2\} \tag{3.2}$$

It now follows from (3.1) and (3.2) that

$$f(m, n) = \{mn/2\}$$

and that $H_{m,n}$ is an m-connected graph on n vertices with as few edges as possible.

We note that since, for any graph G, $\kappa \le \kappa'$ (theorem 3.1), $H_{m,n}$ is also m-edge-connected. Thus, denoting by $g(m, n)$ the least possible number of edges in an m-edge-connected graph on n vertices, we have, for $1 < m < n$

$$g(m, n) = \{mn/2\} \tag{3.3}$$

Exercises

3.3.1 Show that $H_{2r+1,n}$ is $(2r+1)$-connected.

3.3.2 Show that $\kappa(H_{m,n}) = \kappa'(H_{m,n}) = m$.

3.3.3 Find a graph with nine vertices and 23 edges that is 5-connected but not isomorphic to the graph $H_{5,9}$ of figure 3.7*c*.

3.3.4 Show that (3.3) holds for all values of m and n with $m > 1$ and $n > 1$.

FEUE WHEP

5629

3.3.5 Find, for all $\nu \geq 5$, a 2-connected graph G of diameter two with $\varepsilon = 2\nu - 5$.

(Murty, 1969 has shown that every such graph has at least this number of edges.)

REFERENCES

Halin, R. (1969). A theorem on n-connected graphs. *J. Combinatorial Theory*, **7,** 150–54

Harary, F. (1962). The maximum connectivity of a graph. *Proc. Nat. Acad. Sci. U.S.A.*, **48,** 1142–46

Murty, U. S. R. (1969). Extremal nonseparable graphs of diameter 2, in *Proof Techniques in Graph Theory* (ed. F. Harary), Academic Press, New York, pp. 111–18

Whitney, H. (1932). Non-separable and planar graphs. *Trans. Amer. Math. Soc.*, **34,** 339–62

FEUE WHEP

5630

4 Euler Tours and Hamilton Cycles

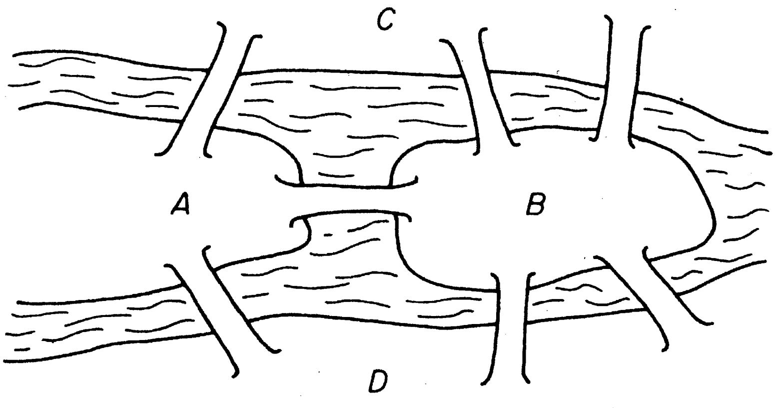
4.1 EULER TOURS

A trail that traverses every edge of G is called an *Euler trail* of G because Euler was the first to investigate the existence of such trails in graphs. In the earliest known paper on graph theory (Euler, 1736), he showed that it was impossible to cross each of the seven bridges of Königsberg once and only once during a walk through the town. A plan of Königsberg and the river Pregel is shown in figure 4.1*a*. As can be seen, proving that such a walk is impossible amounts to showing that the graph of figure 4.1*b* contains no Euler trail.

A *tour* of G is a closed walk that traverses each edge of G at least once. An *Euler tour* is a tour which traverses each edge exactly once (in other words, a closed Euler trail). A graph is *eulerian* if it contains an Euler tour.

Theorem 4.1 A nonempty connected graph is eulerian if and only if it has no vertices of odd degree.

Proof Let G be eulerian, and let C be an Euler tour of G with origin (and terminus) u. Each time a vertex v occurs as an internal vertex of C, two of the edges incident with v are accounted for. Since an Euler tour contains



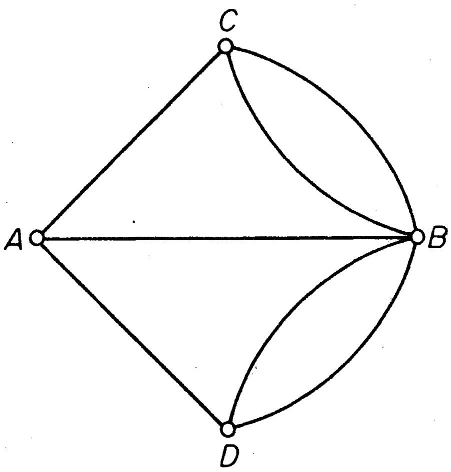


Figure 4.1. The bridges of Königsberg and their graph

5631

every edge of G, $d(v)$ is even for all $v \neq u$. Similarly, since C starts and ends at u, $d(u)$ is also even. Thus G has no vertices of odd degree.

Conversely, suppose that G is a noneulerian connected graph with at least one edge and no vertices of odd degree. Choose such a graph G with as few edges as possible. Since each vertex of G has degree at least two, G contains a closed trail (exercise 1.7.2). Let C be a closed trail of maximum possible length in G. By assumption, C is not an Euler tour of G and so $G-E(C)$ has some component G' with $\varepsilon(G')>0$. Since C is itself eulerian, it has no vertices of odd degree; thus the connected graph G' also has no vertices of odd degree. Since $\varepsilon(G')<\varepsilon(G)$, it follows from the choice of G that G' has an Euler tour C'. Now, because G is connected, there is a vertex v in $V(C)\cap V(C')$, and we may assume, without loss of generality, that v is the origin and terminus of both C and C'. But then CC' is a closed trail of G with $\varepsilon(CC')>\varepsilon(C)$, contradicting the choice of C □

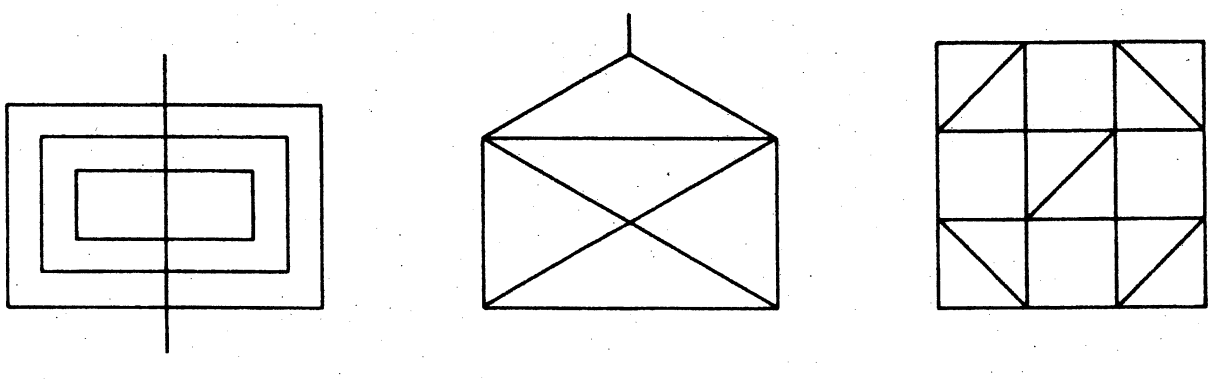
Corollary 4.1 A connected graph has an Euler trail if and only if it has at most two vertices of odd degree.

Proof If G has an Euler trail then, as in the proof of theorem 4.1, each vertex other than the origin and terminus of this trail has even degree.

Conversely, suppose that G is a nontrivial connected graph with at most two vertices of odd degree. If G has no such vertices then, by theorem 4.1, G has a closed Euler trail. Otherwise, G has exactly two vertices, u and v, of odd degree. In this case, let $G+e$ denote the graph obtained from G by the addition of a new edge e joining u and v. Clearly, each vertex of $G+e$ has even degree and so, by theorem 4.1, $G+e$ has an Euler tour $C=v_0e_1v_1\ldots e_{\varepsilon+1}v_{\varepsilon+1}$, where $e_1=e$. The trail $v_1e_2v_2\ldots e_{\varepsilon+1}v_{\varepsilon+1}$ is an Euler trail of G □

Exercises

4.1.1 Which of the following figures can be drawn without lifting one's pen from the paper or covering a line more than once?



4.1.2 If possible, draw an eulerian graph G with ν even and ε odd; otherwise, explain why there is no such graph.

4.1.3 Show that if G is eulerian, then every block of G is eulerian.

FEUE WHEP

5632

4.1.4 Show that if G has no vertices of odd degree, then there are edge-disjoint cycles $C_1, C_2, \ldots, C_m$ such that $E(G) = E(C_1) \cup E(C_2) \cup \ldots \cup E(C_m)$.

4.1.5 Show that if a connected graph G has $2k > 0$ vertices of odd degree, then there are k edge-disjoint trails $Q_1, Q_2, \ldots, Q_k$ in G such that $E(G) = E(Q_1) \cup E(Q_2) \cup \ldots \cup E(Q_k)$.

4.1.6* Let G be nontrivial and eulerian, and let $v \in V$. Show that every trail of G with origin v can be extended to an Euler tour of G if and only if $G - v$ is a forest. (O. Ore)

4.2 HAMILTON CYCLES

A path that contains every vertex of G is called a *Hamilton path* of G; similarly, a *Hamilton cycle* of G is a cycle that contains every vertex of G. Such paths and cycles are named after Hamilton (1856), who described, in a letter to his friend Graves, a mathematical game on the dodecahedron (figure 4.2*a*) in which one person sticks five pins in any five consecutive vertices and the other is required to complete the path so formed to a

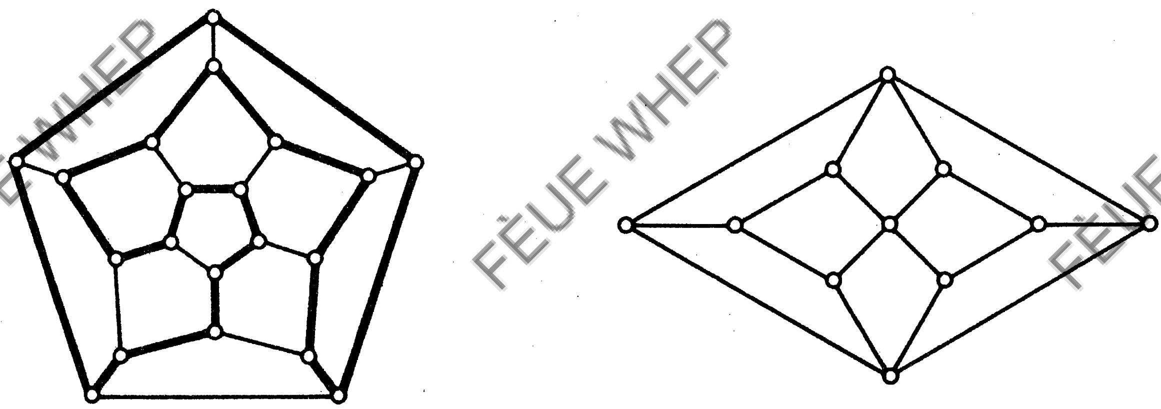


Figure 4.2. (*a*) The dodecahedron; (*b*) the Herschel graph

spanning cycle. A graph is *hamiltonian* if it contains a Hamilton cycle. The dodecahedron is hamiltonian (see figure 4.2*a*); the Herschel graph (figure 4.2*b*) is nonhamiltonian, because it is bipartite and has an odd number of vertices.

In contrast with the case of eulerian graphs, no nontrivial necessary and sufficient condition for a graph to be hamiltonian is known; in fact, the problem of finding such a condition is one of the main unsolved problems of graph theory.

We shall first present a simple, but useful, necessary condition.

Theorem 4.2 If G is hamiltonian then, for every nonempty proper subset S of V

$$\omega(G-S) \leq |S| \tag{4.1}$$

5633

Proof Let C be a Hamilton cycle of G. Then, for every nonempty proper subset S of V

$$\omega(C-S) \leq |S|$$

Also, $C-S$ is a spanning subgraph of $G-S$ and so

$$\omega(G-S) \leq \omega(C-S)$$

The theorem follows □

As an illustration of the above theorem, consider the graph of figure 4.3. This graph has nine vertices; on deleting the three indicated in black, four components remain. Therefore (4.1) is not satisfied and it follows from theorem 4.2 that the graph is nonhamiltonian.

We thus see that theorem 4.2 can sometimes be applied to show that a particular graph is nonhamiltonian. However, this method does not always

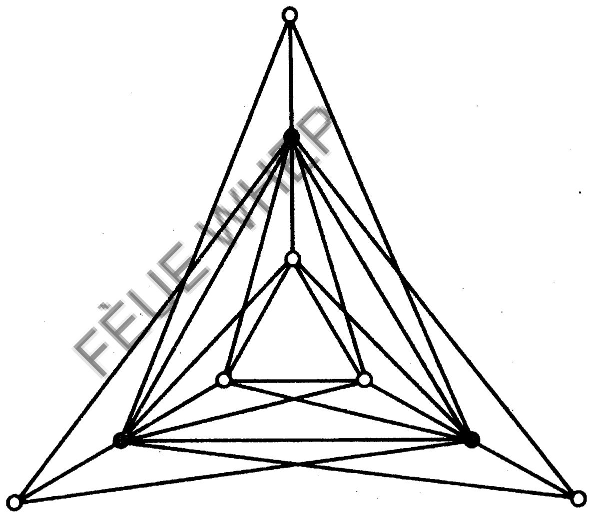


Figure 4.3

work; for instance, the Petersen graph (figure 4.4) is nonhamiltonian, but one cannot deduce this by using theorem 4.2.

We now discuss sufficient conditions for a graph G to be hamiltonian; since a graph is hamiltonian if and only if its underlying simple graph is hamiltonian, it suffices to limit our discussion to simple graphs. We start with a result due to Dirac (1952).

Theorem 4.3 If G is a simple graph with $\nu \geq 3$ and $\delta \geq \nu/2$, then G is hamiltonian.

Proof By contradiction. Suppose that the theorem is false, and let G be a maximal nonhamiltonian simple graph with $\nu \geq 3$ and $\delta \geq \nu/2$. Since $\nu \geq 3$, G cannot be complete. Let u and v be nonadjacent vertices in G. By the choice of G, $G+uv$ is hamiltonian. Moreover, since G is nonhamiltonian,

5634

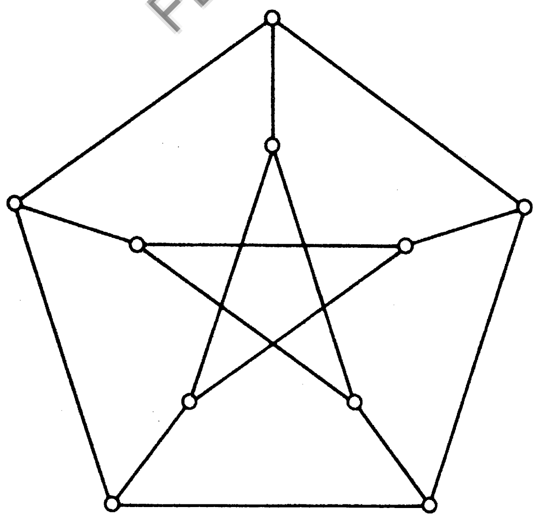


Figure 4.4. The Petersen graph

each Hamilton cycle of $G+uv$ must contain the edge uv. Thus there is a Hamilton path $v_1v_2\ldots v_\nu$ in G with origin $u=v_1$ and terminus $v=v_\nu$. Set

$$S=\{v_i \mid uv_{i+1}\in E\} \quad \text{and} \quad T=\{v_i \mid v_iv\in E\}$$

Since $v_\nu \notin S\cup T$ we have

$$|S\cup T|<\nu \tag{4.2}$$

Furthermore

$$|S\cap T|=0 \tag{4.3}$$

since if $S\cap T$ contained some vertex v_i, then G would have the Hamilton cycle $v_1v_2\ldots v_iv_\nu v_{\nu-1}\ldots v_{i+1}v_1$, contrary to assumption (see figure 4.5).

Using (4.2) and (4.3) we obtain

$$d(u)+d(v)=|S|+|T|=|S\cup T|+|S\cap T|<\nu \tag{4.4}$$

But this contradicts the hypothesis that $\delta\geq\nu/2$ □

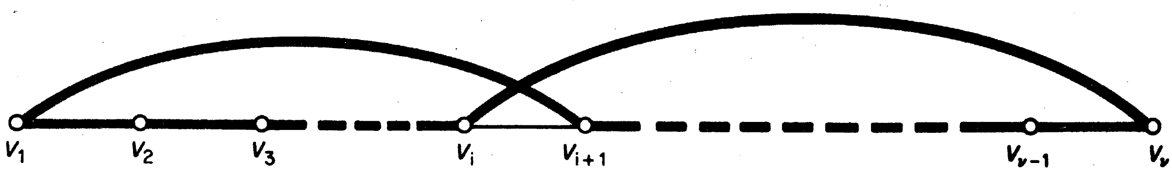


Figure 4.5

Bondy and Chvátal (1974) observed that the proof of theorem 4.3 can be modified to yield stronger sufficient conditions than that obtained by Dirac. The basis of their approach is the following lemma.

Lemma 4.4.1 Let G be a simple graph and let u and v be nonadjacent vertices in G such that

$$d(u)+d(v)\geq\nu \tag{4.5}$$

FÊUE WHEP

5635

Then G is hamiltonian if and only if $G+uv$ is hamiltonian.

Proof If G is hamiltonian then, trivially, so too is $G+uv$. Conversely, suppose that $G+uv$ is hamiltonian but G is not. Then, as in the proof of theorem 4.3, we obtain (4.4). But this contradicts hypothesis (4.5) □

Lemma 4.4.1 motivates the following definition. The *closure* of G is the graph obtained from G by recursively joining pairs of nonadjacent vertices whose degree sum is at least ν until no such pair remains. We denote the closure of G by $c(G)$.

Lemma 4.4.2 $c(G)$ is well defined.

Proof Let G_1 and G_2 be two graphs obtained from G by recursively joining pairs of nonadjacent vertices whose degree sum is at least ν until no such pair remains. Denote by $e_1, e_2, \ldots, e_m$ and $f_1, f_2, \ldots, f_n$ the sequences of edges added to G in obtaining G_1 and G_2, respectively. We shall show that each e_i is an edge of G_2 and each f_j is an edge of G_1.

If possible, let $e_{k+1}=uv$ be the first edge in the sequence $e_1, e_2, \ldots, e_n$ that is not an edge of G_2. Set $H=G+\{e_1, e_2, \ldots, e_k\}$. It follows from the definition of G_1 that

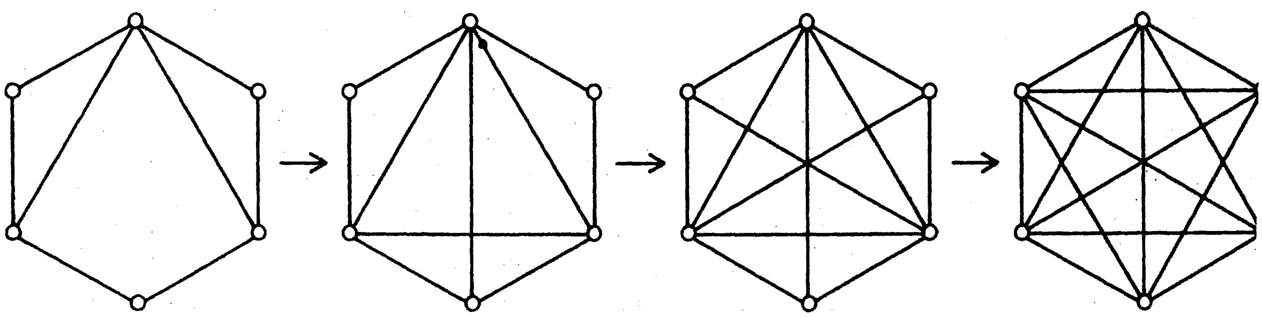
$$d_H(u)+d_H(v)\geq \nu$$

By the choice of e_{k+1}, H is a subgraph of G_2. Therefore

$$d_{G_2}(u)+d_{G_2}(v)\geq \nu$$

This is a contradiction, since u and v are nonadjacent in G_2. Therefore each e_i is an edge of G_2 and, similarly, each f_j is an edge of G_1. Hence $G_1=G_2$, and $c(G)$ is well defined □

Figure 4.6 illustrates the construction of the closure of a graph G on six vertices. It so happens that in this example $c(G)$ is complete; note, however, that this is by no means always the case.



G $c(G)$

Figure 4.6. The closure of a graph

5636

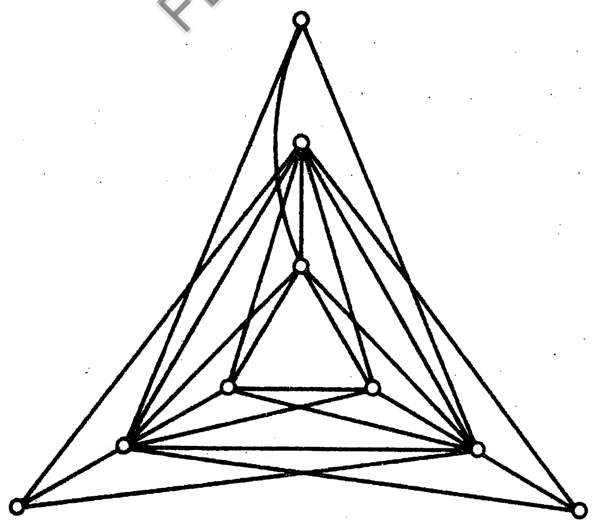


Figure 4.7. A hamiltonian graph

Theorem 4.4 A simple graph is hamiltonian if and only if its closure is hamiltonian.

Proof Apply lemma 4.4.1 each time an edge is added in the formation of the closure □

Theorem 4.4 has a number of interesting consequences. First, upon making the trivial observation that all complete graphs on at least three vertices are hamiltonian, we obtain the following result.

Corollary 4.4 Let G be a simple graph with $\nu \geq 3$. If $c(G)$ is complete, then G is hamiltonian.

Consider, for example, the graph of figure 4.7. One readily checks that its closure is complete. Therefore, by corollary 4.4, it is hamiltonian. It is perhaps interesting to note that the graph of figure 4.7 can be obtained from the graph of figure 4.3 by altering just one end of one edge, and yet we have results (corollary 4.4 and theorem 4.2) which tell us that this one is hamiltonian whereas the other is not.

Corollary 4.4 can be used to deduce various sufficient conditions for a graph to be hamiltonian in terms of its vertex degrees. For example, since $c(G)$ is clearly complete when $\delta \geq \nu/2$, Dirac's condition (theorem 4.3) is an immediate corollary. A more general condition than that of Dirac was obtained by Chvátal (1972).

Theorem 4.5 Let G be a simple graph with degree sequence $(d_1, d_2, \ldots, d_\nu)$, where $d_1 \leq d_2 \leq \ldots \leq d_\nu$ and $\nu \geq 3$. Suppose that there is no value of m less than $\nu/2$ for which $d_m \leq m$ and $d_{\nu-m} < \nu - m$. Then G is hamiltonian.

5637

Proof Let G satisfy the hypothesis of the theorem. We shall show that its closure $c(G)$ is complete, and the conclusion will then follow from corollary 4.4. We denote the degree of a vertex v in $c(G)$ by $d'(v)$.

Assume that $c(G)$ is not complete, and let u and v be two nonadjacent vertices in $c(G)$ with

$$d'(u) \leq d'(v) \tag{4.6}$$

and $d'(u)+d'(v)$ as large as possible; since no two nonadjacent vertices in $c(G)$ can have degree sum ν or more, we have

$$d'(u)+d'(v) < \nu \tag{4.7}$$

Now denote by S the set of vertices in $V\backslash\{v\}$ which are nonadjacent to v in $c(G)$, and by T the set of vertices in $V\backslash\{u\}$ which are nonadjacent to u in $c(G)$. Clearly

$$|S| = \nu - 1 - d'(v) \quad \text{and} \quad |T| = \nu - 1 - d'(u) \tag{4.8}$$

Furthermore, by the choice of u and v, each vertex in S has degree at most $d'(u)$ and each vertex in $T \cup \{u\}$ has degree at most $d'(v)$. Setting $d'(u) = m$ and using (4.7) and (4.8), we find that $c(G)$ has at least m vertices of degree at most m and at least $\nu - m$ vertices of degree less than $\nu - m$. Because G is a spanning subgraph of $c(G)$, the same is true of G; therefore $d_m \leq m$ and $d_{\nu-m} < \nu - m$. But this is contrary to hypothesis since, by (4.6) and (4.7), $m < \nu/2$. We conclude that $c(G)$ is indeed complete and hence, by corollary 4.4, that G is hamiltonian. □

One can often deduce that a given graph is hamiltonian simply by computing its degree sequence and applying theorem 4.5. This method works with the graph of figure 4.7 but not with the graph G of figure 4.6, even though the closure of the latter graph is complete. From these examples, we see that theorem 4.5 is stronger than theorem 4.3 but not as strong as corollary 4.4.

A sequence of real numbers $(p_1, p_2, \ldots, p_n)$ is said to be *majorised* by another such sequence $(q_1, q_2, \ldots, q_n)$ if $p_i \leq q_i$ for $1 \leq i \leq n$. A graph G is *degree-majorised* by a graph H if $\nu(G) = \nu(H)$ and the nondecreasing degree sequence of G is majorised by that of H. For instance, the 5-cycle is degree-majorised by $K_{2,3}$ because (2, 2, 2, 2, 2) is majorised by (2, 2, 2, 3, 3). The family of degree-maximal nonhamiltonian graphs (those that are degree-majorised by no others) admits of a simple description. We first introduce the notion of the join of two graphs. The *join* $G \vee H$ of disjoint graphs G and H is the graph obtained from $G + H$ by joining each vertex of G to each vertex of H; it is represented diagrammatically as in figure 4.8.

Now, for $1 \leq m < n/2$, let $C_{m,n}$ denote the graph $K_m \vee (K_m^c + K_{n-2m})$, depicted in figure 4.9$a$; two specific examples, $C_{1,5}$ and $C_{2,5}$, are shown in figures 4.9b and 4.9c.

5638

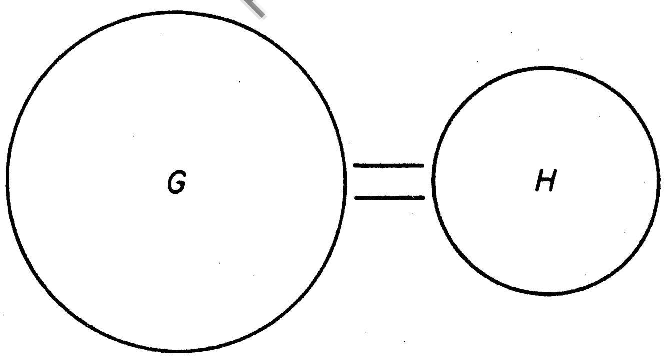


Figure 4.8. The join of G and H

That $C_{m,n}$ is nonhamiltonian follows immediately from theorem 4.2; for if S denotes the set of m vertices of degree $n-1$ in $C_{m,n}$, we have $\omega(C_{m,n}-S)=m+1>|S|$.

Theorem 4.6 (Chvátal, 1972) If G is a nonhamiltonian simple graph with $\nu\geq 3$, then G is degree-majorised by some $C_{m,\nu}$.

Proof Let G be a nonhamiltonian simple graph with degree sequence $(d_1, d_2, \ldots, d_\nu)$, where $d_1\leq d_2\leq\ldots\leq d_\nu$ and $\nu\geq 3$. Then, by theorem 4.5, there exists $m<\nu/2$ such that $d_m\leq m$ and $d_{\nu-m}<\nu-m$. Therefore $(d_1, d_2, \ldots, d_\nu)$ is majorised by the sequence

$$(m, \ldots, m, \nu-m-1, \ldots, \nu-m-1, \nu-1, \ldots, \nu-1)$$

with m terms equal to m, $\nu-2m$ terms equal to $\nu-m-1$ and m terms equal to $\nu-1$, and this latter sequence is the degree sequence of $C_{m,\nu}$ □

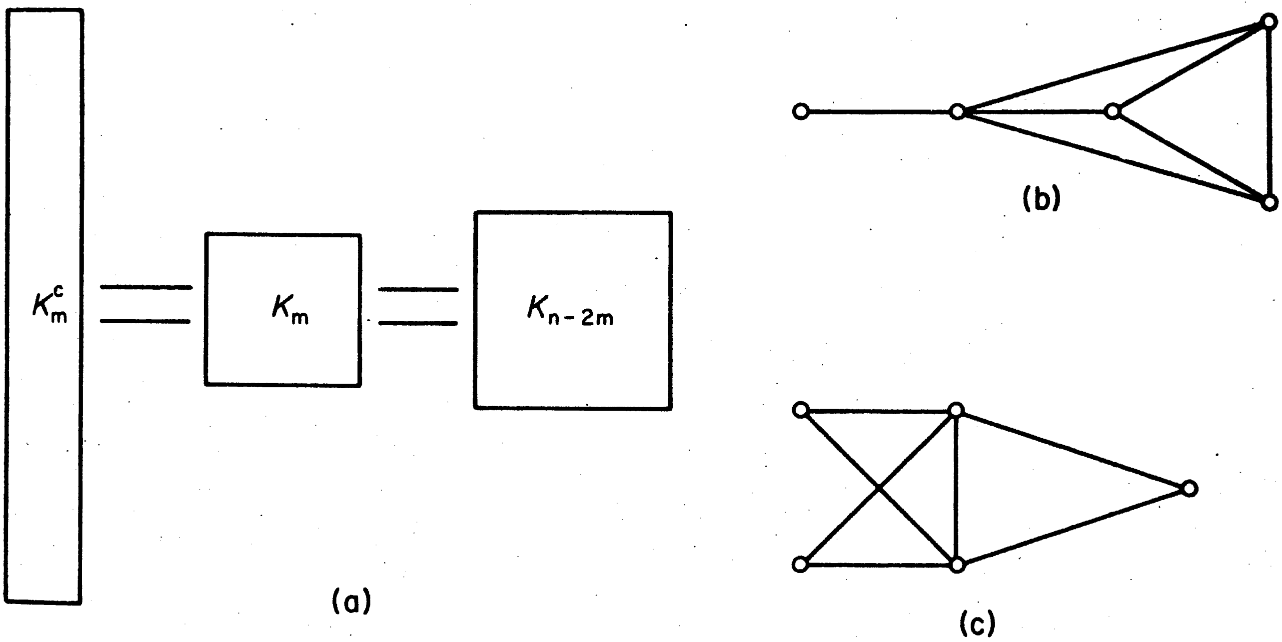


Figure 4.9. (*a*) $C_{m,n}$; (*b*) $C_{1,5}$; (*c*) $C_{2,5}$

FEUE WHEP

5639

From theorem 4.6 we can deduce a result due to Ore (1961) and Bondy (1972).

Corollary 4.6 If G is a simple graph with $\nu \geq 3$ and $\varepsilon > \binom{\nu-1}{2}+1$, then G is hamiltonian. Moreover, the only nonhamiltonian simple graphs with ν vertices and $\binom{\nu-1}{2}+1$ edges are $C_{1,\nu}$ and, for $\nu = 5$, $C_{2,5}$.

Proof Let G be a nonhamiltonian simple graph with $\nu \geq 3$. By theorem 4.6, G is degree-majorised by $C_{m,\nu}$ for some positive integer $m < \nu/2$. Therefore, by theorem 1.1,

$$\varepsilon(G) \leq \varepsilon(C_{m,\nu}) \tag{4.9}$$

$$= \tfrac{1}{2}(m^2 + (\nu-2m)(\nu-m-1) + m(\nu-1))$$

$$= \binom{\nu-1}{2} + 1 - \tfrac{1}{2}(m-1)(m-2) - (m-1)(\nu-2m-1)$$

$$\leq \binom{\nu-1}{2} + 1 \tag{4.10}$$

Furthermore, equality can only hold in (4.9) if G has the same degree sequence as $C_{m,\nu}$; and equality can only hold in (4.10) if either $m = 2$ and $\nu = 5$, or $m = 1$. Hence $\varepsilon(G)$ can equal $\binom{\nu-1}{2}+1$ only if G has the same degree sequence as $C_{1,\nu}$ or $C_{2,5}$, which is easily seen to imply that $G \cong C_{1,\nu}$ or $G \cong C_{2,5}$ □

Exercises

4.2.1 Show that if either
(*a*) G is not 2-connected, or
(*b*) G is bipartite with bipartition (X, Y) where $|X| \neq |Y|$,
then G is nonhamiltonian.

4.2.2 A mouse eats his way through a $3\times3\times3$ cube of cheese by tunnelling through all of the 27 $1\times1\times1$ subcubes. If he starts at one corner and always moves on to an uneaten subcube, can he finish at the centre of the cube?

4.2.3 Show that if G has a Hamilton path then, for every proper subset S of V, $\omega(G-S) \leq |S|+1$.

4.2.4* Let G be a nontrivial simple graph with degree sequence $(d_1, d_2, \ldots, d_\nu)$, where $d_1 \leq d_2 \leq \ldots \leq d_\nu$. Show that if there is no

5640

value of m less than $(\nu+1)/2$ for which $d_m < m$ and $d_{\nu-m+1} < \nu - m$, then G has a Hamilton path. (V. Chvátal)

4.2.5 (*a*) Let G be a simple graph with degree sequence $(d_1, d_2, \ldots, d_\nu)$ and let G^c have degree sequence $(d'_1, d'_2, \ldots, d'_\nu)$ where $d_1 \leq d_2 \leq \ldots \leq d_\nu$ and $d'_1 \leq d'_2 \leq \ldots \leq d'_\nu$. Show that if $d_m \geq d'_m$ for all $m \leq \nu/2$, then G has a Hamilton path.

(*b*) Deduce that if G is self-complementary, then G has a Hamilton path. (C. R. J. Clapham)

4.2.6* Let G be a simple bipartite graph with bipartition (X, Y), where $|X| = |Y| \geq 2$, and let G have degree sequence $(d_1, d_2, \ldots, d_\nu)$, where $d_1 \leq d_2 \leq \ldots \leq d_\nu$. Show that if there is no value of m less than or equal to $\nu/4$ for which $d_m \leq m$ and $d_{\nu/2} \leq \nu/2 - m$, then G is hamiltonian. (V. Chvátal)

4.2.7 Prove corollary 4.6 directly from corollary 4.4.

4.2.8 Show that if G is simple with $\nu \geq 6\delta$ and $\varepsilon > \binom{\nu-\delta}{2} + \delta^2$, then G is hamiltonian. (P. Erdös)

4.2.9* Show that if G is a connected graph with $\nu > 2\delta$, then G has a path of length at least 2δ. (G. A. Dirac)

(Dirac, 1952 has also shown that if G is a 2-connected simple graph with $\nu \geq 2\delta$, then G has a cycle of length at least 2δ.)

4.2.10 Using the remark to exercise 4.2.9, show that every $2k$-regular simple graph on $4k+1$ vertices is hamiltonian $(k \geq 1)$. (C. St. J. A. Nash-Williams)

4.2.11 G is *Hamilton-connected* if every two vertices of G are connected by a Hamilton path.

(*a*) Show that if G is Hamilton-connected and $\nu \geq 4$, then $\varepsilon \geq [\frac{1}{2}(3\nu+1)]$.

(*b*)* For $\nu \geq 4$, construct a Hamilton-connected graph G with $\varepsilon = [\frac{1}{2}(3\nu+1)]$. (J. W. Moon)

4.2.12 G is *hypohamiltonian* if G is not hamiltonian but $G - v$ is hamiltonian for every $v \in V$. Show that the Petersen graph (figure 4.4) is hypohamiltonian.

(Herz, Duby and Vigué, 1967 have shown that it is, in fact, the smallest such graph.)

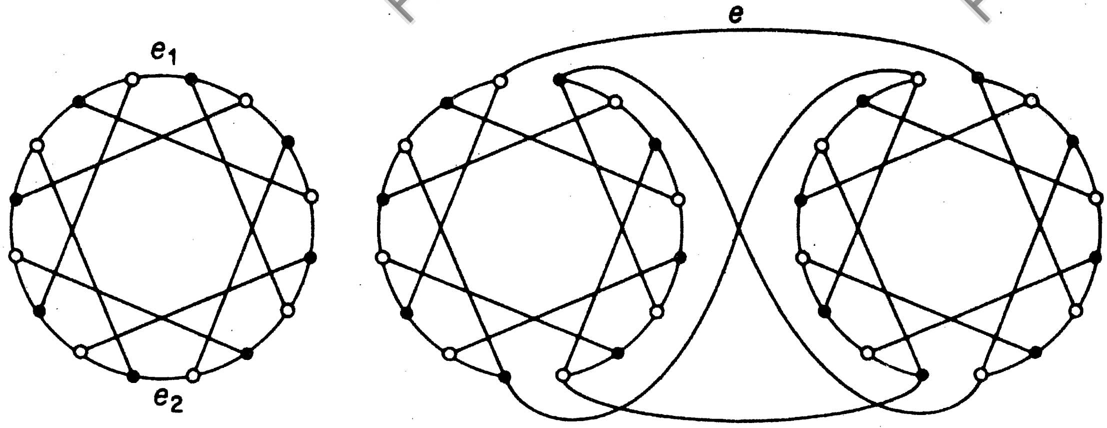
4.2.13* G is *hypotraceable* if G has no Hamilton path but $G - v$ has a Hamilton path for every $v \in V$. Show that the Thomassen graph (p. 240) is hypotraceable.

4.2.14 (*a*) Show that there is no Hamilton cycle in the graph G_1 below which contains exactly one of the edges e_1 and e_2.

(*b*) Using (*a*), show that every Hamilton cycle in G_2 includes the edge e.

(*c*) Deduce that the Horton graph (p. 240) is nonhamiltonian.

5641



G_1 G_2

4.2.15 Describe a good algorithm for
(*a*) constructing the closure of a graph;
(*b*) finding a Hamilton cycle if the closure is complete.

APPLICATIONS

4.3 THE CHINESE POSTMAN PROBLEM

In his job, a postman picks up mail at the post office, delivers it, and then returns to the post office. He must, of course, cover each street in his area at least once. Subject to this condition, he wishes to choose his route in such a way that he walks as little as possible. This problem is known as the *Chinese postman problem*, since it was first considered by a Chinese mathematician, Kuan (1962).

In a weighted graph, we define the *weight* of a tour $v_0e_1v_1\ldots e_nv_0$ to be $\sum_{i=1}^{n} w(e_i)$. Clearly, the Chinese postman problem is just that of finding a minimum-weight tour in a weighted connected graph with non-negative weights. We shall refer to such a tour as an *optimal tour*.

If G is eulerian, then any Euler tour of G is an optimal tour because an Euler tour is a tour that traverses each edge exactly once. The Chinese postman problem is easily solved in this case, since there exists a good algorithm for determining an Euler tour in an eulerian graph. The algorithm, due to Fleury (see Lucas, 1921), constructs an Euler tour by tracing out a trail, subject to the one condition that, at any stage, a cut edge of the untraced subgraph is taken only if there is no alternative.

Fleury's Algorithm

1. Choose an arbitrary vertex v_0, and set $W_0 = v_0$.
2. Suppose that the trail $W_i = v_0e_1v_1\ldots e_iv_i$ has been chosen.

5642

Then choose an edge e_{i+1} from $E\backslash\{e_1, e_2, \ldots, e_i\}$ in such a way that
(i) e_{i+1} is incident with v_i;
(ii) unless there is no alternative, e_{i+1} is not a cut edge of

$$G_i = G - \{e_1, e_2, \ldots, e_i\}$$

3. Stop when step 2 can no longer be implemented.

By its definition, Fleury's algorithm constructs a trail in G.

Theorem 4.7 If G is eulerian, then any trail in G constructed by Fleury's algorithm is an Euler tour of G.

Proof Let G be eulerian, and let $W_n = v_0e_1v_1 \ldots e_nv_n$ be a trail in G constructed by Fleury's algorithm. Clearly, the terminus v_n must be of degree zero in G_n. It follows that $v_n = v_0$; in other words, W_n is a closed trail.

Suppose, now, that W_n is not an Euler tour of G, and let S be the set of vertices of positive degree in G_n. Then S is nonempty and $v_n \in \bar{S}$, where $\bar{S} = V\backslash S$. Let m be the largest integer such that $v_m \in S$ and $v_{m+1} \in \bar{S}$. Since W_n terminates in $\bar{S}$, e_{m+1} is the only edge of $[S, \bar{S}]$ in G_m, and hence is a cut edge of G_m (see figure 4.10).

Let e be any other edge of G_m incident with v_m. It follows (step 2) that e must also be a cut edge of G_m, and hence of $G_m[S]$. But since $G_m[S] = G_n[S]$, every vertex in $G_m[S]$ is of even degree. However, this implies (exercise 2.2.6*a*) that $G_m[S]$ has no cut edge, a contradiction □

The proof that Fleury's algorithm is a good algorithm is left as an exercise (exercise 4.3.2).

If G is not eulerian, then any tour in G and, in particular, an optimal tour in G, traverses some edges more than once. For example, in the graph of figure 4.11*a* *xuywvzwyxuwvxzyx* is an optimal tour (exercise 4.3.1). Notice that the four edges *ux*, *xy*, *yw* and *wv* are traversed twice by this tour.

It is convenient, at this stage, to introduce the operation of duplication of an edge. An edge e is said to be *duplicated* when its ends are joined by a

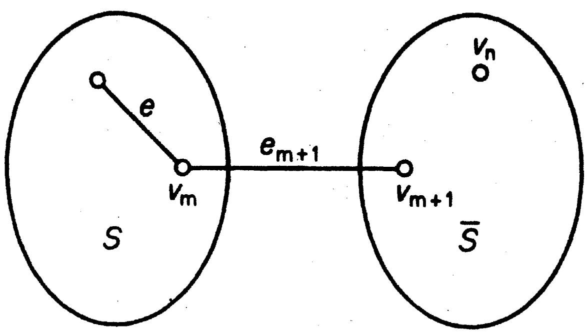


Figure 4.10

5643

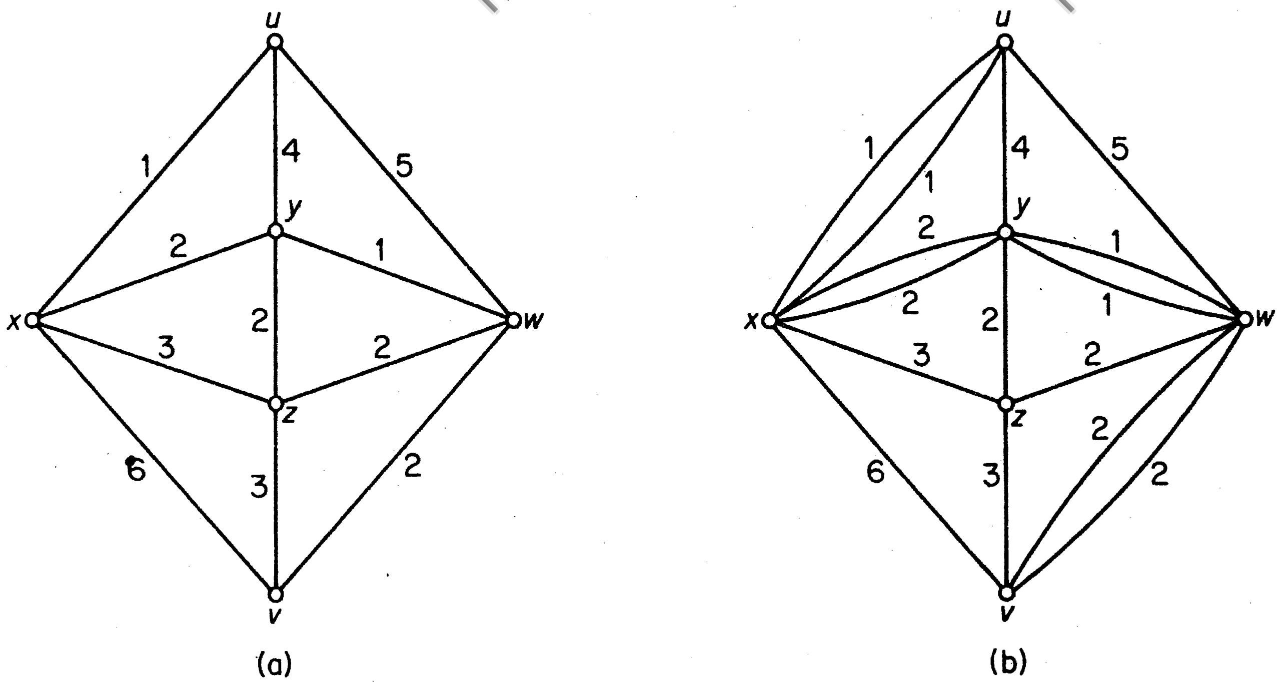


Figure 4.11

new edge of weight $w(e)$. By duplicating the edges ux, xy, yw and wv in the graph of figure 4.11*a*, we obtain the graph shown in figure 4.11*b*.

We may now rephrase the Chinese postman problem as follows: given a weighted graph G with non-negative weights,

(i) find, by duplicating edges, an eulerian weighted supergraph G^* of G such that $\sum_{e\in E(G^*)\setminus E(G)} w(e)$ is as small as possible;

(ii) find an Euler tour in G^*.

That this is equivalent to the Chinese postman problem follows from the observation that a tour of G in which edge e is traversed $m(e)$ times corresponds to an Euler tour in the graph obtained from G by duplicating e $m(e)-1$ times, and vice versa.

We have already presented a good algorithm for solving (ii), namely Fleury's algorithm. A good algorithm for solving (i) has been given by Edmonds and Johnson (1973). Unfortunately, it is too involved to be presented here. However, we shall consider one special case which affords an easy solution. This is the case where G has exactly two vertices of odd degree.

Suppose that G has exactly two vertices u and v of odd degree; let G^* be an eulerian spanning supergraph of G obtained by duplicating edges, and write E^* for $E(G^*)$. Clearly the subgraph $G^*[E^*\setminus E]$ of G^* (induced by the edges of G^* that are not in G) also has only the two vertices u and v of odd degree. It follows from corollary 1.1 that u and v are in the same component of $G^*[E^*\setminus E]$ and hence that they are connected by a (u, v)-path P^*.

5644

Clearly

$$\sum_{e \in E^* \setminus E} w(e) \geq w(P^*) \geq w(P)$$

where P is a minimum-weight (u, v)-path in G. Thus $\sum_{e \in E^* \setminus E} w(e)$ is a minimum when G^* is obtained from G by duplicating each of the edges on a minimum-weight (u, v)-path. A good algorithm for finding such a path was given in section 1.8.

Exercises

4.3.1 Show that $xuywvzwyxuwvxzyx$ is an optimal tour in the weighted graph of figure 4.11*a*.

4.3.2 Draw a flow diagram summarising Fleury's algorithm, and show that it is a good algorithm.

4.4 THE TRAVELLING SALESMAN PROBLEM

A travelling salesman wishes to visit a number of towns and then return to his starting point. Given the travelling times between towns, how should he plan his itinerary so that he visits each town exactly once and travels in all for as short a time as possible? This is known as the *travelling salesman problem.* In graphical terms, the aim is to find a minimum-weight Hamilton cycle in a weighted complete graph. We shall call such a cycle an *optimal cycle.* In contrast with the shortest path problem and the connector problem, no efficient algorithm for solving the travelling salesman problem is known. It is therefore desirable to have a method for obtaining a reasonably good (but not necessarily optimal) solution. We shall show how some of our previous theory can be employed to this end.

One possible approach is to first find a Hamilton cycle C, and then search for another of smaller weight by suitably modifying C. Perhaps the simplest such modification is as follows.

Let $C = v_1 v_2 \ldots v_\nu v_1$. Then, for all i and j such that $1 < i+1 < j < \nu$, we can obtain a new Hamilton cycle

$$C_{ij} = v_1 v_2 \ldots v_i v_j v_{j-1} \ldots v_{i+1} v_{j+1} v_{j+2} \ldots v_\nu v_1$$

by deleting the edges $v_i v_{i+1}$ and $v_j v_{j+1}$ and adding the edges $v_i v_j$ and $v_{i+1} v_{j+1}$, as shown in figure 4.12.

If, for some i and j

$$w(v_i v_j) + w(v_{i+1} v_{j+1}) < w(v_i v_{i+1}) + w(v_j v_{j+1})$$

the cycle C_{ij} will be an improvement on C.

After performing a sequence of the above modifications, one is left with a cycle that can be improved no more by these methods. This final cycle will

5645

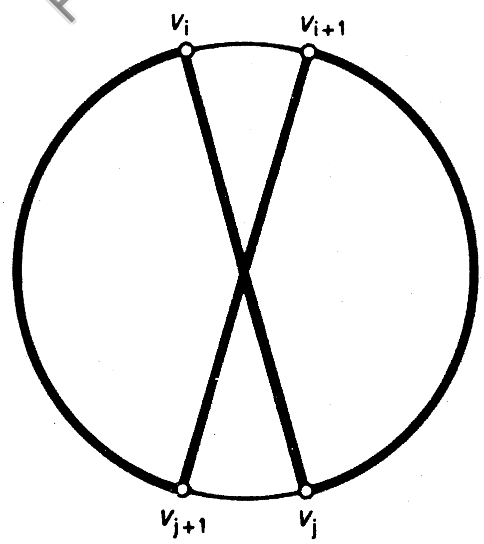


Figure 4.12

almost certainly not be optimal, but it is a reasonable assumption that it will often be fairly good; for greater accuracy, the procedure can be repeated several times, starting with a different cycle each time.

As an example, consider the weighted graph shown in figure 4.13; it is the same graph as was used in our illustration of Kruskal's algorithm in section 2.5.

Starting with the cycle L MC NY Pa Pe T L, we can apply a sequence of three modifications, as illustrated in figure 4.14, and end up with the cycle L NY MC T Pe Pa L of weight 192.

An indication of how good our solution is can sometimes be obtained by applying Kruskal's algorithm. Suppose that C is an optimal cycle in G. Then, for any vertex v, $C-v$ is a Hamilton path in $G-v$, and is therefore a

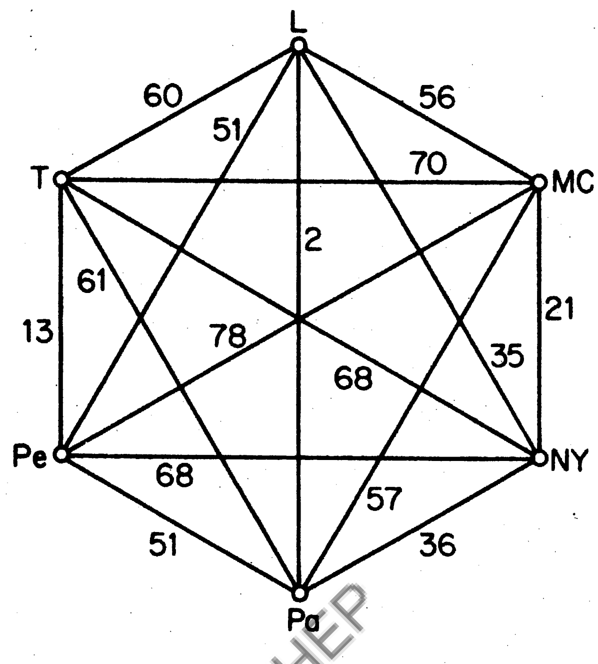


Figure 4.13

FEUE WHEP

5646

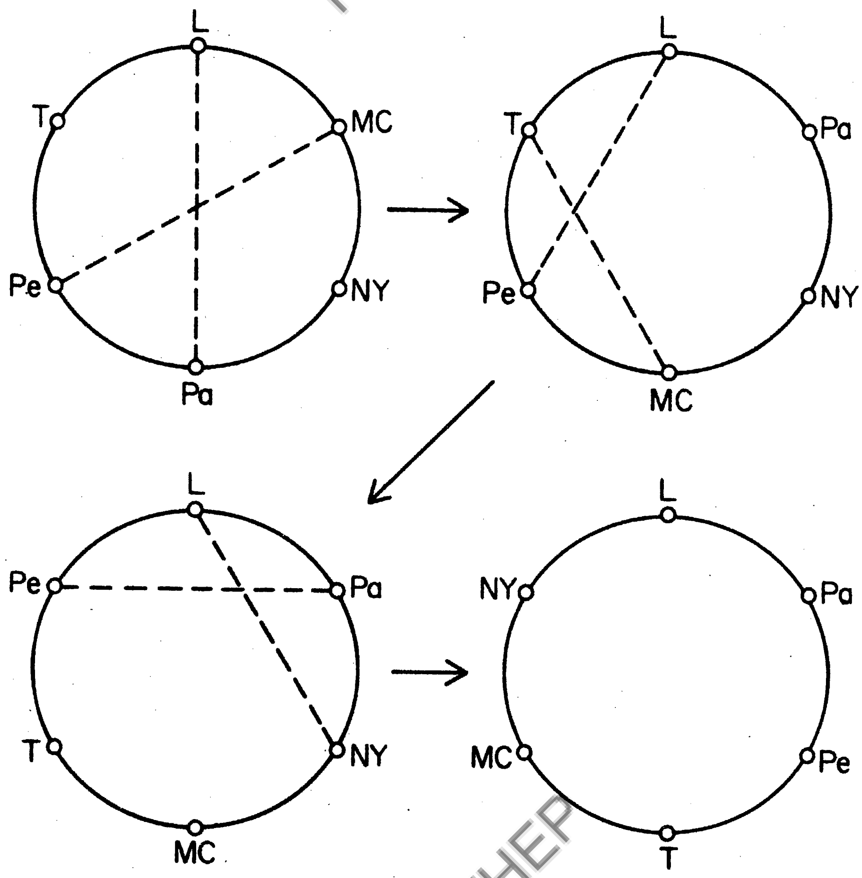


Figure 4.14

spanning tree of $G - v$. It follows that if T is an optimal tree in $G - v$, and if e and f are two edges incident with v such that $w(e) + w(f)$ is as small as possible, then $w(T) + w(e) + w(f)$ will be a lower bound on $w(C)$. In our example, taking NY as the vertex v, we find (see figure 4.15) that

$$w(T) = 122 \qquad w(e) = 21 \quad \text{and} \quad w(f) = 35$$

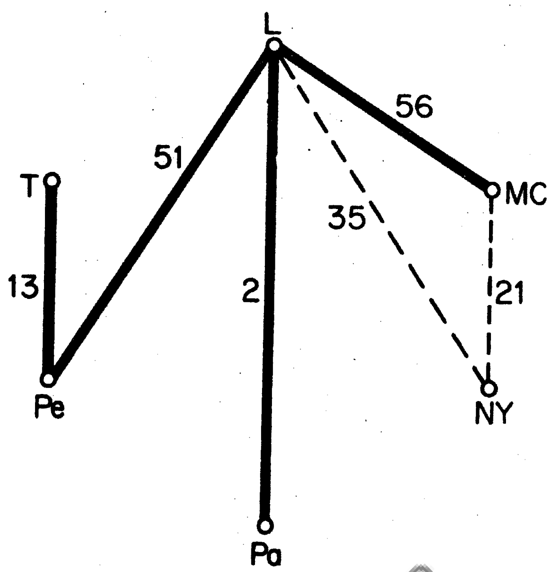


Figure 4.15

5647

We can therefore conclude that the weight $w(C)$ of an optimal cycle in the graph of figure 4.13 satisfies

$$178 \le w(C) \le 192$$

The methods described here have been further developed by Lin (1965) and Held and Karp (1970; 1971). In particular, Lin has found that the cycle modification procedure can be made more efficient by replacing three edges at a time rather than just two; somewhat surprisingly, however, it is not advantageous to extend this same idea further. For a survey of the travelling salesman problem, see Bellmore and Nemhauser (1968).

Exercise

4.4.1* Let G be a weighted complete graph in which the weights satisfy the triangle inequality: $w(xy)+w(yz) \ge w(xz)$ for all x, y, $z \in V$. Show that an optimal cycle in G has weight at most $2w(T)$, where T is an optimal tree in G.

(D. J. Rosencrantz, R. E. Stearns, P. M. Lewis)

REFERENCES

Bellmore, M. and Nemhauser, G. L. (1968). The traveling salesman problem: a survey. *Operations Res.*, **16,** 538–58

Bondy, J. A. (1972). Variations on the hamiltonian theme. *Canad. Math. Bull.*, **15,** 57–62

Bondy, J. A. and Chvátal, V. (1974). A method in graph theory (in press)

Chvátal, V. (1972). On Hamilton's ideals. *J. Combinatorial Theory* B, **12,** 163–68

Dirac, G. A. (1952). Some theorems on abstract graphs. *Proc. London Math. Soc.*, **2,** 69–81

Edmonds, J. and Johnson, E. L. (1973). Matching, Euler tours and the Chinese postman. *Math. Programming*, **5,** 88–124

Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis. *Comment. Academiae Sci. I. Petropolitanae*, **8,** 128–40

Hamilton, W. R. (1856). Letter to John T. Graves on the Icosian, 17 Oct., 1856, in *The Mathematical Papers of Sir William Rowan Hamilton* (eds. H. Halberstam and R. E. Ingram), vol. 3 (Algebra), Cambridge University Press, 1931, pp. 612–25.

Held, M. and Karp, R. M. (1970). The traveling-salesman problem and minimum spanning trees. *Operations Res.*, **18,** 1138–62

Held, M. and Karp, R. M. (1971). The traveling-salesman problem and minimum spanning trees: part II, *Math. Programming*, **1,** 6–25

Herz, J. C., Duby, J. J. and Vigué, F. (1967). Recherche systématique des

5648

graphes hypohamiltoniens, in *Théorie des Graphes* (ed. P. Rosenstiehl), Dunod-Gordon and Breach, pp. 153–59

Kuan, M-K. (1962). Graphic programming using odd or even points. *Chinese Math.*, **1,** 273–77

Lin, S. (1965). Computer solutions of the traveling salesman problem, *Bell System Tech. J.*, **44,** 2245–69

Lucas, E. (1921). *Récréations Mathématiques IV*, Paris

Ore, O. (1961). Arc coverings of graphs. *Ann. Mat. Pura Appl.*, **55,** 315–21

5649

5 Matchings

5.1 MATCHINGS

A subset M of E is called a *matching* in G if its elements are links and no two are adjacent in G; the two ends of an edge in M are said to be *matched under M*. A matching M *saturates* a vertex v, and v is said to be *M-saturated*, if some edge of M is incident with v; otherwise, v is *M-unsaturated*. If every vertex of G is M-saturated, the matching M is *perfect*. M is a *maximum matching* if G has no matching M' with $|M'|>|M|$; clearly, every perfect matching is maximum. Maximum and perfect matchings in graphs are indicated in figure 5.1.

Let M be a matching in G. An *M-alternating path* in G is a path whose edges are alternately in $E\backslash M$ and M. For example, the path $v_5v_8v_1v_7v_6$ in the graph of figure 5.1*a* is an M-alternating path. An *M-augmenting path* is an M-alternating path whose origin and terminus are M-unsaturated.

Theorem 5.1 (Berge, 1957) A matching M in G is a maximum matching if and only if G contains no M-augmenting path.

Proof Let M be a matching in G, and suppose that G contains an M-augmenting path $v_0v_1 \ldots v_{2m+1}$. Define $M' \subseteq E$ by

$$M' = (M\backslash\{v_1v_2, v_3v_4, \ldots, v_{2m-1}v_{2m}\}) \cup \{v_0v_1, v_2v_3, \ldots, v_{2m}v_{2m+1}\}$$

Then M' is a matching in G, and $|M'|=|M|+1$. Thus M is not a maximum matching.

Conversely, suppose that M is not a maximum matching, and let M' be a maximum matching in G. Then

$$|M'|>|M| \tag{5.1}$$

Set $H=G[M\,\Delta M']$, where $M\,\Delta M'$ denotes the symmetric difference of M and M' (see figure 5.2).

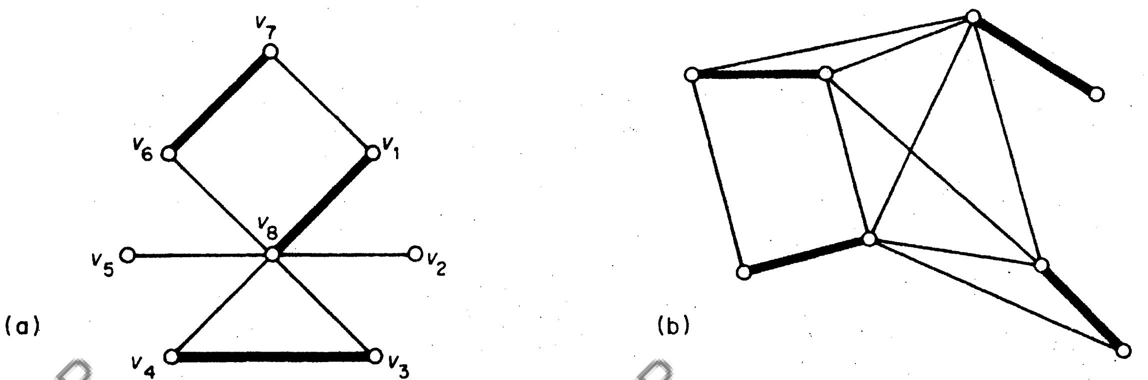


Figure 5.1. (*a*) A maximum matching; (*b*) a perfect matching

FÉUE WHEP

5650

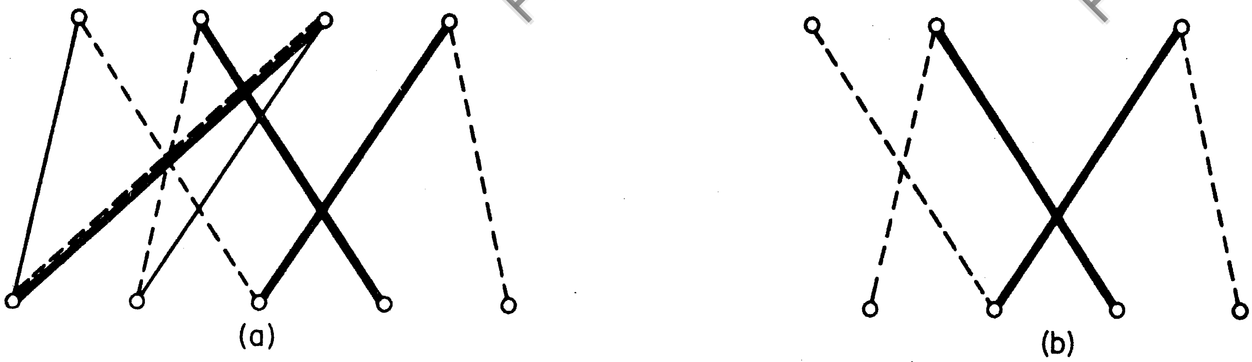


Figure 5.2. (*a*) G, with M heavy and M' broken; (*b*) $G[M \Delta M']$

Each vertex of H has degree either one or two in H, since it can be incident with at most one edge of M and one edge of M'. Thus each component of H is either an even cycle with edges alternately in M and M', or else a path with edges alternately in M and M'. By (5.1), H contains more edges of M' than of M, and therefore some path component P of H must start and end with edges of M'. The origin and terminus of P, being M'-saturated in H, are M-unsaturated in G. Thus P is an M-augmenting path in G □

Exercises

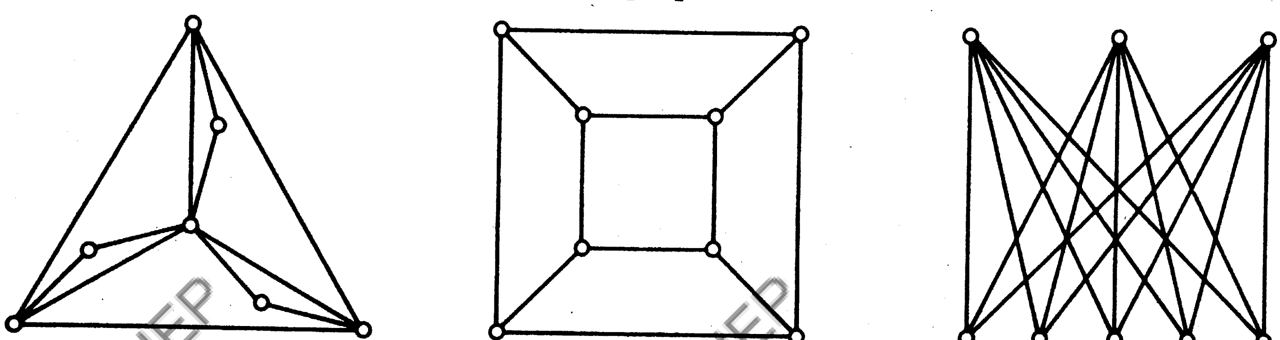
5.1.1 (*a*) Show that every k-cube has a perfect matching ($k \geq 2$).
(*b*) Find the number of different perfect matchings in K_{2n} and $K_{n,n}$.

5.1.2 Show that a tree has at most one perfect matching.

5.1.3 For each $k > 1$, find an example of a k-regular simple graph that has no perfect matching.

5.1.4 Two people play a game on a graph G by alternately selecting distinct vertices $v_0, v_1, v_2, \ldots$ such that, for $i > 0$, v_i is adjacent to v_{i-1}. The last player able to select a vertex wins. Show that the first player has a winning strategy if and only if G has no perfect matching.

5.1.5 A *k-factor* of G is a k-regular spanning subgraph of G, and G is *k-factorable* if there are edge-disjoint k-factors $H_1, H_2, \ldots, H_n$ such that $G = H_1 \cup H_2 \cup \ldots \cup H_n$.
(*a*)* Show that
(i) $K_{n,n}$ and K_{2n} are 1-factorable;
(ii) the Petersen graph is not 1-factorable.
(*b*) Which of the following graphs have 2-factors?



5651

(*c*) Using Dirac's theorem (4.3), show that if G is simple, with ν even and $\delta \geq (\nu/2)+1$, then G has a 3-factor.

5.1.6* Show that K_{2n+1} can be expressed as the union of n connected 2-factors ($n \geq 1$).

5.2 MATCHINGS AND COVERINGS IN BIPARTITE GRAPHS

For any set S of vertices in G, we define the *neighbour set* of S in G to be the set of all vertices adjacent to vertices in S; this set is denoted by $N_G(S)$. Suppose, now, that G is a bipartite graph with bipartition (X, Y). In many applications one wishes to find a matching of G that saturates every vertex in X; an example is the personnel assignment problem, to be discussed in section 5.4. Necessary and sufficient conditions for the existence of such a matching were first given by Hall (1935).

Theorem 5.2 Let G be a bipartite graph with bipartition (X, Y). Then G contains a matching that saturates every vertex in X if and only if

$$|N(S)| \geq |S| \quad \text{for all} \quad S \subseteq X \tag{5.2}$$

Proof Suppose that G contains a matching M which saturates every vertex in X, and let S be a subset of X. Since the vertices in S are matched under M with distinct vertices in $N(S)$, we clearly have $|N(S)| \geq |S|$.

Conversely, suppose that G is a bipartite graph satisfying (5.2), but that G contains no matching saturating all the vertices in X. We shall obtain a contradiction. Let M^* be a maximum matching in G. By our supposition, M^* does not saturate all vertices in X. Let u be an M^*-unsaturated vertex in X, and let Z denote the set of all vertices connected to u by M^*-alternating paths. Since M^* is a maximum matching, it follows from theorem 5.1 that u is the only M^*-unsaturated vertex in Z. Set $S = Z \cap X$ and $T = Z \cap Y$ (see figure 5.3).

Clearly, the vertices in $S \backslash \{u\}$ are matched under M^* with the vertices in T. Therefore

$$|T| = |S| - 1 \tag{5.3}$$

and $N(S) \supseteq T$. In fact, we have

$$N(S) = T \tag{5.4}$$

since every vertex in $N(S)$ is connected to u by an M^*-alternating path. But

5652

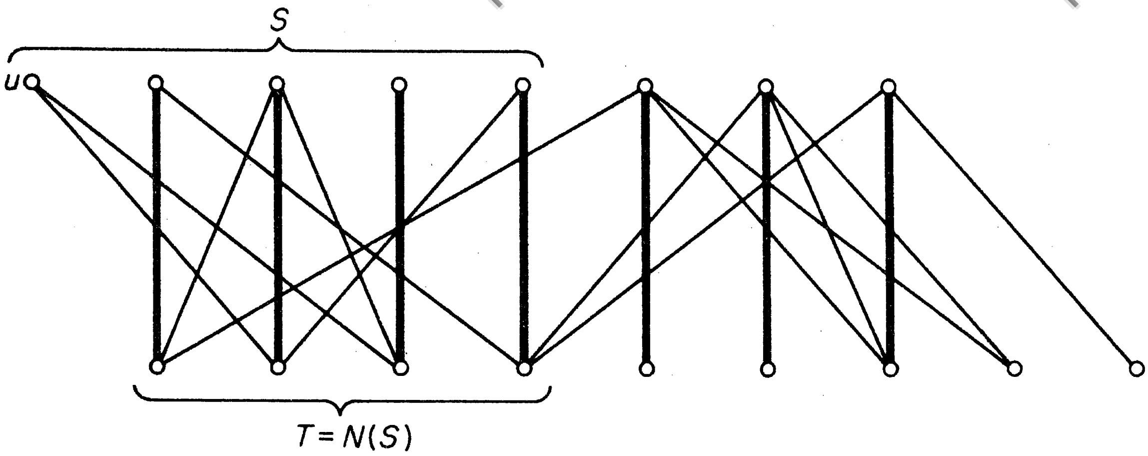


Figure 5.3

(5.3) and (5.4) imply that

$$|N(S)| = |S| - 1 < |S|$$

contradicting assumption (5.2) □

The above proof provides the basis of a good algorithm for finding a maximum matching in a bipartite graph. This algorithm will be presented in section 5.4.

Corollary 5.2 If G is a k-regular bipartite graph with $k > 0$, then G has a perfect matching.

Proof Let G be a k-regular bipartite graph with bipartition (X, Y). Since G is k-regular, $k\,|X| = |E| = k\,|Y|$ and so, since $k > 0$, $|X| = |Y|$. Now let S be a subset of X and denote by E_1 and E_2 the sets of edges incident with vertices in S and $N(S)$, respectively. By definition of $N(S)$, $E_1 \subseteq E_2$ and therefore

$$k\,|N(S)| = |E_2| \geq |E_1| = k\,|S|$$

It follows that $|N(S)| \geq |S|$ and hence, by theorem 5.2, that G has a matching M saturating every vertex in X. Since $|X| = |Y|$, M is a perfect matching □

Corollary 5.2 is sometimes known as the *marriage theorem*, since it can be more colourfully restated as follows: if every girl in a village knows exactly k boys, and every boy knows exactly k girls, then each girl can marry a boy she knows, and each boy can marry a girl he knows.

A *covering* of a graph G is a subset K of V such that every edge of G has at least one end in K. A covering K is a *minimum covering* if G has no covering K' with $|K'| < |K|$ (see figure 5.4).

If K is a covering of G, and M is a matching of G, then K contains at

FEUE WHEP

5653

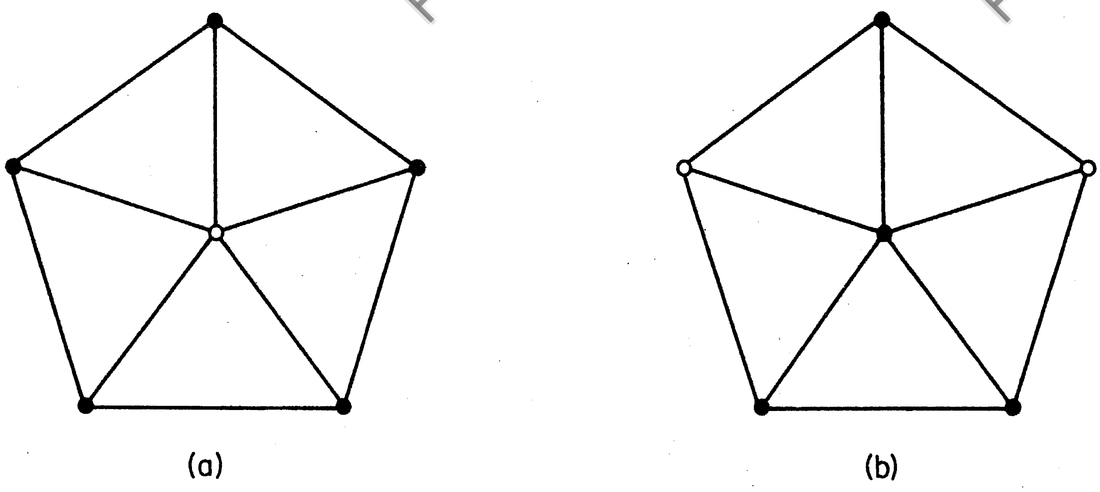


Figure 5.4. (*a*) A covering; (*b*) a minimum covering

least one end of each of the edges in M. Thus, for any matching M and any covering K, $|M| \leq |K|$. Indeed, if M^* is a maximum matching and $\tilde{K}$ is a minimum covering, then

$$|M^*| \leq |\tilde{K}| \tag{5.5}$$

In general, equality does not hold in (5.5) (see, for example, figure 5.4). However, if G is bipartite we do have $|M^*| = |\tilde{K}|$. This result, due to König (1931), is closely related to Hall's theorem. Before presenting its proof, we make a simple, but important, observation.

Lemma 5.3 Let M be a matching and K be a covering such that $|M| = |K|$. Then M is a maximum matching and K is a minimum covering.

Proof If M^* is a maximum matching and $\tilde{K}$ is a minimum covering then, by (5.5),

$$|M| \leq |M^*| \leq |\tilde{K}| \leq |K|$$

Since $|M| = |K|$, it follows that $|M| = |M^*|$ and $|K| = |\tilde{K}|$ □

Theorem 5.3 In a bipartite graph, the number of edges in a maximum matching is equal to the number of vertices in a minimum covering.

Proof Let G be a bipartite graph with bipartition (X, Y), and let M^* be a maximum matching of G. Denote by U the set of M^*-unsaturated vertices in X, and by Z the set of all vertices connected by M^*-alternating paths to vertices of U. Set $S = Z \cap X$ and $T = Z \cap Y$. Then, as in the proof of theorem 5.2, we have that every vertex in T is M^*-saturated and $N(S) = T$. Define $\tilde{K} = (X \backslash S) \cup T$ (see figure 5.5). Every edge of G must have at least one of its ends in $\tilde{K}$. For, otherwise, there would be an edge with one end in

5654

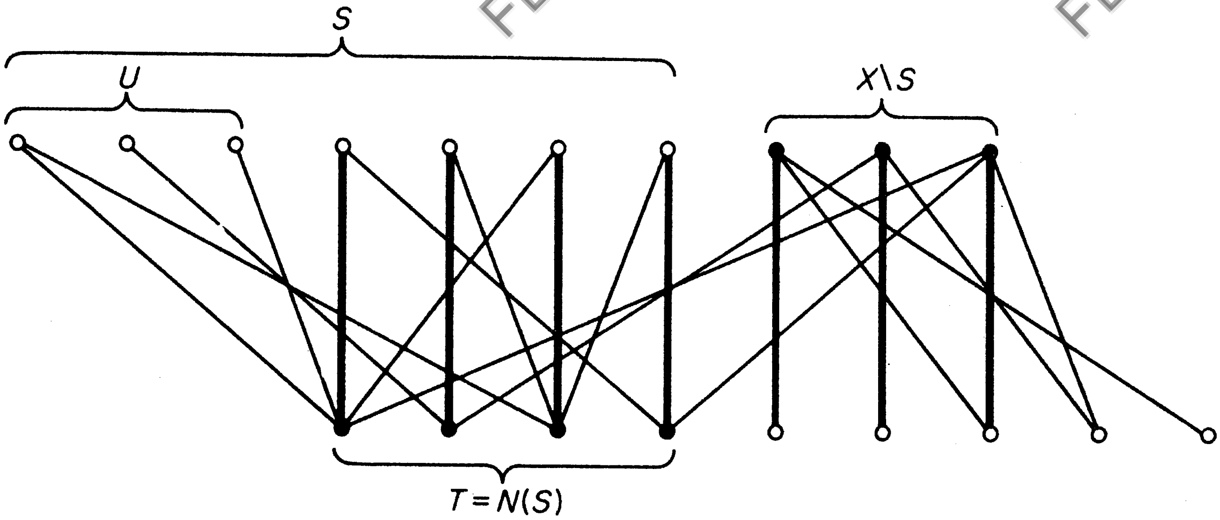


Figure 5.5

S and one end in $Y\backslash T$, contradicting $N(S)=T$. Thus $\tilde{K}$ is a covering of G and clearly

$$|M^*|=|\tilde{K}|$$

By lemma 5.3, $\tilde{K}$ is a minimum covering, and the theorem follows □

Exercises

5.2.1 Show that it is impossible, using 1×2 rectangles, to exactly cover an 8×8 square from which two opposite 1×1 corner squares have been removed.

5.2.2 (*a*) Show that a bipartite graph G has a perfect matching if and only if $|N(S)|\geq|S|$ for all $S\subseteq V$.

(*b*) Give an example to show that the above statement does not remain valid if the condition that G be bipartite is dropped.

5.2.3 For $k>0$, show that

(*a*) every k-regular bipartite graph is 1-factorable;

(*b*)* every $2k$-regular graph is 2-factorable. (J. Petersen)

5.2.4 Let $A_1, A_2, \ldots, A_m$ be subsets of a set S. A *system of distinct representatives* for the family $(A_1, A_2, \ldots, A_m)$ is a subset $\{a_1, a_2, \ldots, a_m\}$ of S such that $a_i\in A_i$, $1\leq i\leq m$, and $a_i\neq a_j$ for $i\neq j$. Show that $(A_1, A_2, \ldots, A_m)$ has a system of distinct representatives if and only if $\left|\bigcup_{i\in J} A_i\right|\geq|J|$ for all subsets J of $\{1, 2, \ldots, m\}$. (P. Hall)

5.2.5 A *line* of a matrix is a row or a column of the matrix. Show that the minimum number of lines containing all the 1's of a (0, 1)-matrix is equal to the maximum number of 1's, no two of which are in the same line.

5655

5.2.6 (*a*) Prove the following generalisation of Hall's theorem (5.2): if G is a bipartite graph with bipartition (X, Y), the number of edges in a maximum matching of G is

$$|X| - \max_{S \subseteq X} \{|S| - |N(S)|\}$$

(D. König, O. Ore)

(*b*) Deduce that if G is simple with $|X| = |Y| = n$ and $\varepsilon > (k-1)n$, then G has a matching of cardinality k.

5.2.7 Deduce Hall's theorem (5.2) from König's theorem (5.3).

5.2.8* A non-negative real matrix $\mathbf{Q}$ is *doubly stochastic* if the sum of the entries in each row of $\mathbf{Q}$ is 1 and the sum of the entries in each column of $\mathbf{Q}$ is 1. A *permutation matrix* is a (0, 1)-matrix which has exactly one 1 in each row and each column. (Thus every permutation matrix is doubly stochastic.) Show that

(*a*) every doubly stochastic matrix is necessarily square;

(*b*) every doubly stochastic matrix $\mathbf{Q}$ can be expressed as a convex linear combination of permutation matrices; that is

$$\mathbf{Q} = c_1\mathbf{P}_1 + c_2\mathbf{P}_2 + \ldots + c_k\mathbf{P}_k$$

where each $\mathbf{P}_i$ is a permutation matrix, each c_i is a non-negative real number, and $\sum_1^k c_i = 1$. (G. Birkhoff, J. von Neumann)

5.2.9 Let H be a finite group and let K be a subgroup of H. Show that there exist elements $h_1, h_2, \ldots, h_n \in H$ such that $h_1K, h_2K, \ldots, h_nK$ are the left cosets of K and $Kh_1, Kh_2, \ldots, Kh_n$ are the right cosets of K. (P. Hall)

5.3 PERFECT MATCHINGS

A necessary and sufficient condition for a graph to have a perfect matching was obtained by Tutte (1947). The proof given here is due to Lovász (1973).

A component of a graph is *odd* or *even* according as it has an odd or even number of vertices. We denote by $o(G)$ the number of odd components of G.

Theorem 5.4 G has a perfect matching if and only if

$$o(G-S) \leq |S| \quad \text{for all} \quad S \subset V \tag{5.6}$$

Proof It clearly suffices to prove the theorem for simple graphs.

Suppose first that G has a perfect matching M. Let S be a proper subset of V, and let $G_1, G_2, \ldots, G_n$ be the odd components of $G-S$. Because G_i is odd, some vertex u_i of G_i must be matched under M with a vertex v_i of S (see figure 5.6). Therefore, since $\{v_1, v_2, \ldots, v_n\} \subseteq S$

$$o(G-S) = n = |\{v_1, v_2, \ldots, v_n\}| \leq |S|$$

5656

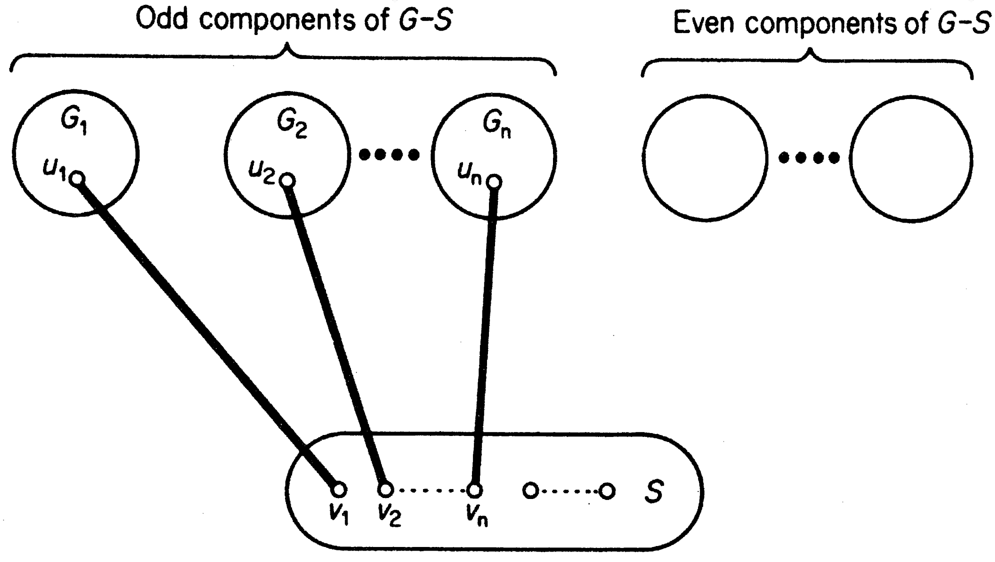


Figure 5.6

Conversely, suppose that G satisfies (5.6) but has no perfect matching. Then G is a spanning subgraph of a maximal graph G^* having no perfect matching. Since $G-S$ is a spanning subgraph of G^*-S we have $o(G^*-S) \leq o(G-S)$ and so, by (5.6),

$$o(G^*-S) \leq |S| \quad \text{for all} \quad S \subset V(G^*) \tag{5.7}$$

In particular, setting $S=\emptyset$, we see that $o(G^*)=0$, and so $\nu(G^*)$ is even.

Denote by U the set of vertices of degree $\nu-1$ in G^*. Since G^* clearly has a perfect matching if $U=V$, we may assume that $U \neq V$. We shall show that G^*-U is a disjoint union of complete graphs. Suppose, to the contrary, that some component of G^*-U is not complete. Then, in this component, there are vertices x, y and z such that $xy \in E(G^*)$, $yz \in E(G^*)$ and $xz \notin E(G^*)$ (exercise 1.6.14). Moreover, since $y \notin U$, there is a vertex w in G^*-U such that $yw \notin E(G^*)$. The situation is illustrated in figure 5.7.

Since G^* is a maximal graph containing no perfect matching, G^*+e has a perfect matching for all $e \notin E(G^*)$. Let M_1 and M_2 be perfect matchings in G^*+xz and G^*+yw, respectively, and denote by H the subgraph of

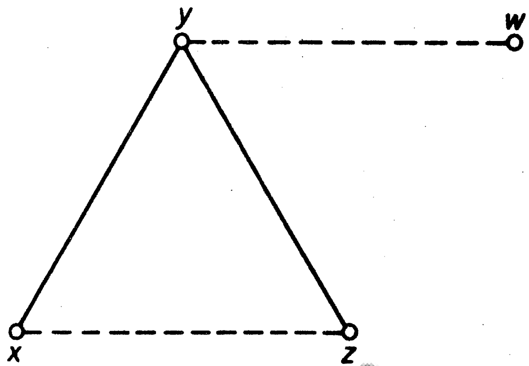


Figure 5.7

FEUE WHEP

5657

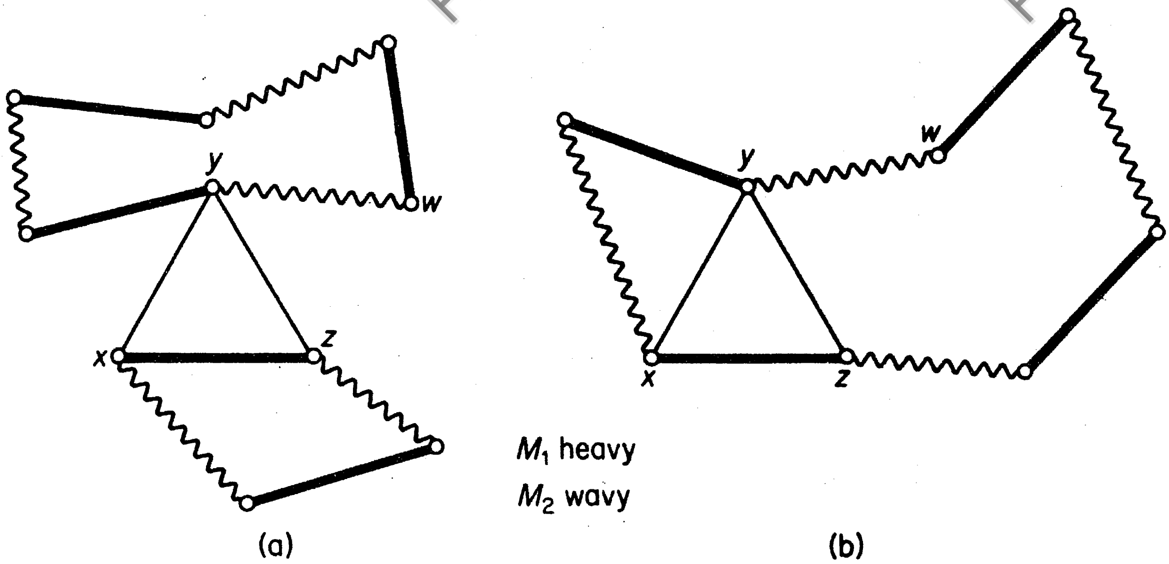


Figure 5.8

$G^* \cup \{xz, yw\}$ induced by $M_1 \Delta M_2$. Since each vertex of H has degree two, H is a disjoint union of cycles. Furthermore, all of these cycles are even, since edges of M_1 alternate with edges of M_2 around them. We distinguish two cases:

Case 1 xz and yw are in different components of H (figure 5.8a). Then, if yw is in the cycle C of H, the edges of M_1 in C, together with the edges of M_2 not in C, constitute a perfect matching in G^*, contradicting the definition of G^*.

Case 2 xz and yw are in the same component C of H. By symmetry of x and z, we may assume that the vertices x, y, w and z occur in that order on C (figure 5.8b). Then the edges of M_1 in the section $yw \ldots z$ of C, together with the edge yz and the edges of M_2 not in the section $yw \ldots z$ of C,

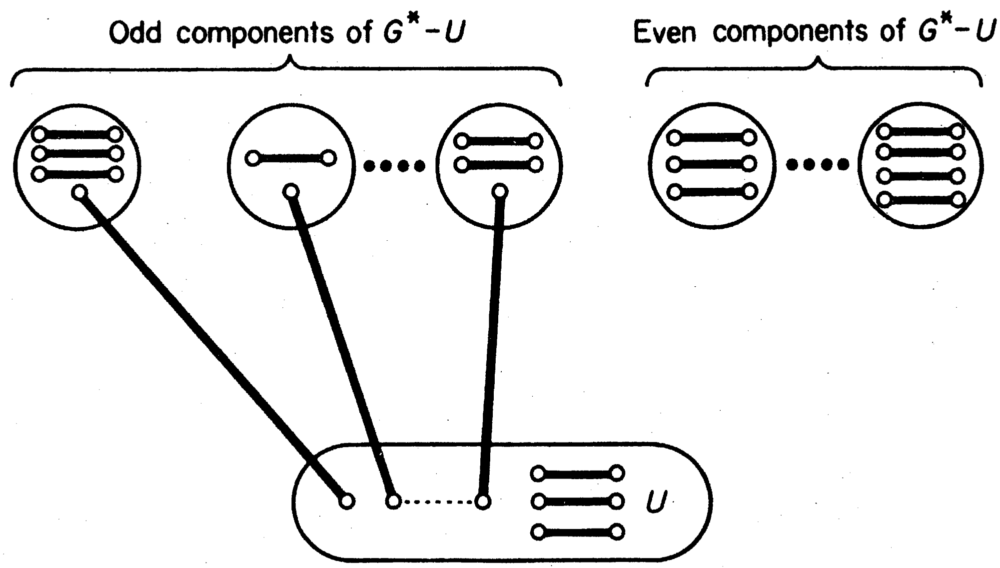


Figure 5.9

FEUE WHEP

5658

constitute a perfect matching in G^*, again contradicting the definition of G^*.

Since both case 1 and case 2 lead to contradictions, it follows that G^*-U is indeed a disjoint union of complete graphs.

Now, by (5.7), $o(G^*-U) \leq |U|$. Thus at most $|U|$ of the components of G^*-U are odd. But then G^* clearly has a perfect matching: one vertex in each odd component of G^*-U is matched with a vertex of U; the remaining vertices in U, and in components of G^*-U, are then matched as indicated in figure 5.9.

Since G^* was assumed to have no perfect matching we have obtained the desired contradiction. Thus G does indeed have a perfect matching □

The above theorem can also be proved with the aid of Hall's theorem (see Anderson, 1971).

From Tutte's theorem, we now deduce a result first obtained by Petersen (1891).

Corollary 5.4 Every 3-regular graph without cut edges has a perfect matching.

Proof Let G be a 3-regular graph without cut edges, and let S be a proper subset of V. Denote by $G_1, G_2, \ldots, G_n$ the odd components of $G-S$, and let m_i be the number of edges with one end in G_i and one end in S, $1 \leq i \leq n$. Since G is 3-regular

$$\sum_{v \in V(G_i)} d(v) = 3\nu(G_i) \quad \text{for} \quad 1 \leq i \leq n \tag{5.8}$$

and

$$\sum_{v \in S} d(v) = 3|S| \tag{5.9}$$

By (5.8), $m_i = \sum_{v \in V(G_i)} d(v) - 2\varepsilon(G_i)$ is odd. Now $m_i \neq 1$ since G has no cut edge. Thus

$$m_i \geq 3 \quad \text{for} \quad 1 \leq i \leq n \tag{5.10}$$

It follows from (5.10) and (5.9) that

$$o(G-S) = n \leq \frac{1}{3}\sum_{i=1}^{n} m_i \leq \frac{1}{3}\sum_{v \in S} d(v) = |S|$$

Therefore, by theorem 5.4, G has a perfect matching □

A 3-regular graph with cut edges need not have a perfect matching. For example, it follows from theorem 5.4 that the graph G of figure 5.10 has no perfect matching, since $o(G-v) = 3$.

FEUE WHEP

5659

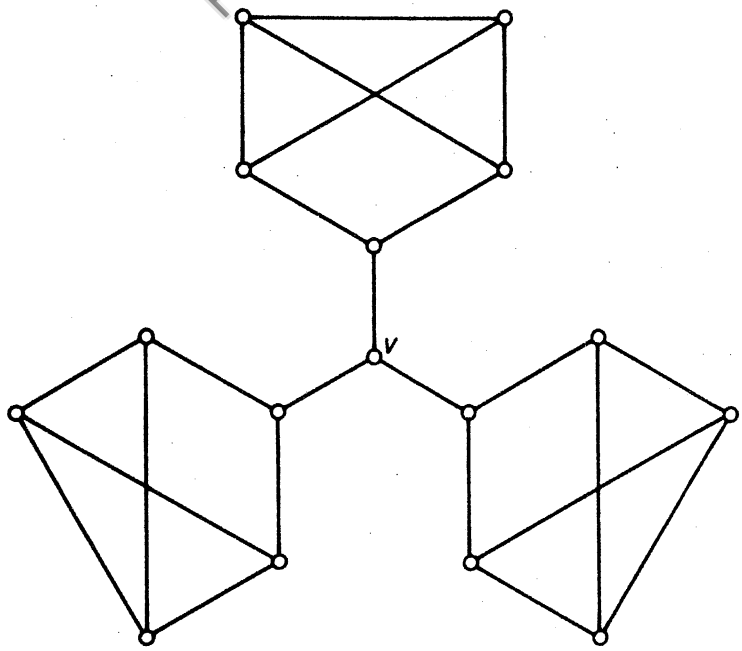


Figure 5.10

Exercises

5.3.1* Derive Hall's theorem (5.2) from Tutte's theorem (5.4).

5.3.2 Prove the following generalisation of corollary 5.4: if G is a $(k-1)$-edge-connected k-regular graph with ν even, then G has a perfect matching.

5.3.3 Show that a tree G has a perfect matching if and only if $o(G-v)=1$ for all $v \in V$. (V. Chungphaisan)

5.3.4* Prove the following generalisation of Tutte's theorem (5.4): the number of edges in a maximum matching of G is $\frac{1}{2}(\nu - d)$, where $d = \max_{S \subset V}\{o(G-S)-|S|\}$. (C. Berge)

5.3.5 (*a*) Using Tutte's theorem (5.4), characterise the maximal simple graphs which have no perfect matching.

(*b*) Let G be simple, with ν even and $\delta < \nu/2$. Show that if $\varepsilon > \binom{\delta}{2} + \binom{\nu-2\delta-1}{2} + \delta(\nu-\delta)$, then G has a perfect matching.

APPLICATIONS

5.4 THE PERSONNEL ASSIGNMENT PROBLEM

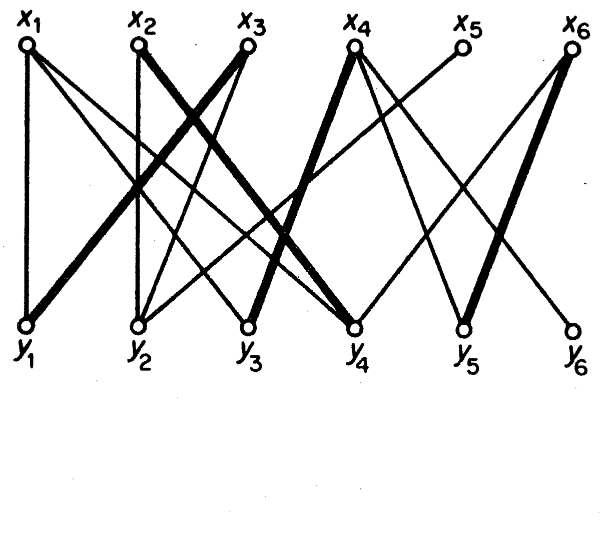
In a certain company, n workers $X_1, X_2, \ldots, X_n$ are available for n jobs $Y_1, Y_2, \ldots, Y_n$, each worker being qualified for one or more of these jobs. Can all the men be assigned, one man per job, to jobs for which they are qualified? This is the *personnel assignment problem*.

5660

We construct a bipartite graph G with bipartition (X, Y), where $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$, and x_i is joined to y_j if and only if worker X_i is qualified for job Y_j. The problem becomes one of determining whether or not G has a perfect matching. According to Hall's theorem (5.2), either G has such a matching or there is a subset S of X such that $|N(S)| < |S|$. In the sequel, we shall present an algorithm to solve the personnel assignment problem. Given any bipartite graph G with bipartition (X, Y), the algorithm either finds a matching of G that saturates every vertex in X or, failing this, finds a subset S of X such that $|N(S)| < |S|$.

The basic idea behind the algorithm is very simple. We start with an arbitrary matching M. If M saturates every vertex in X, then it is a matching of the required type. If not, we choose an M-unsaturated vertex u in X and systematically search for an M-augmenting path with origin u. Our method of search, to be described in detail below, finds such a path P if one exists; in this case $\hat{M} = M \,\Delta E(P)$ is a larger matching than M, and hence saturates more vertices in X. We then repeat the procedure with $\hat{M}$ instead of M. If such a path does not exist, the set Z of all vertices which are connected to u by M-alternating paths is found. Then (as in the proof of theorem 5.2) $S = Z \cap X$ satisfies $|N(S)| < |S|$.

Let M be a matching in G, and let u be an M-unsaturated vertex in X. A tree $H \subseteq G$ is called an *M-alternating tree rooted at u* if (i) $u \in V(H)$, and (ii) for every vertex v of H, the unique (u, v)-path in H is an M-alternating path. An M-alternating tree in a graph is shown in figure 5.11.



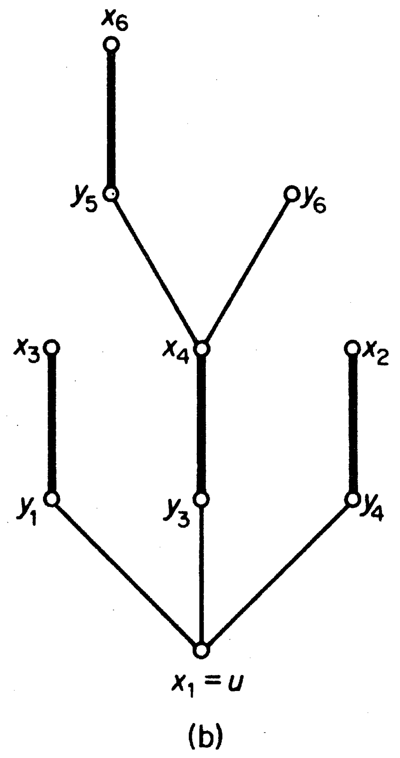


Figure 5.11. (*a*) A matching M in G; (*b*) an M-alternating tree in G

FEUE WHEP

5661

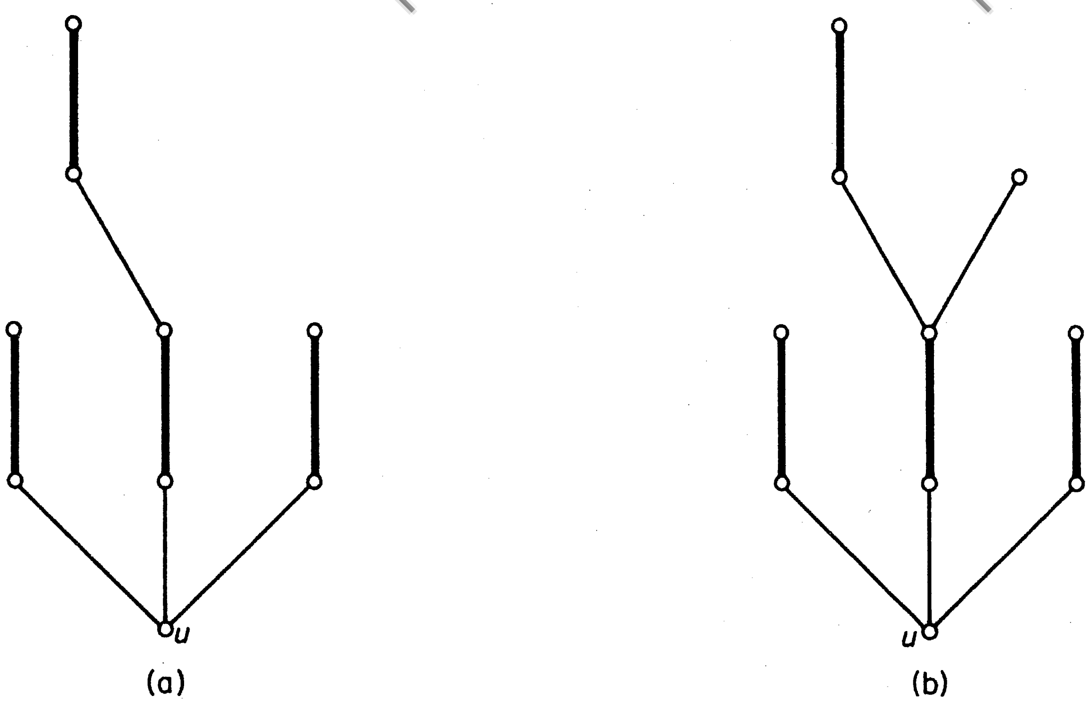


Figure 5.12. (*a*) Case (i); (*b*) case (ii)

The search for an M-augmenting path with origin u involves 'growing' an M-alternating tree H rooted at u. This procedure was first suggested by Edmonds (1965). Initially, H consists of just the single vertex u. It is then grown in such a way that, at any stage, either

(i) all vertices of H except u are M-saturated and matched under M (as in figure 5.12a), or
(ii) H contains an M-unsaturated vertex different from u (as in figure 5.12b).

If (i) is the case (as it is initially) then, setting $S = V(H) \cap X$ and $T = V(H) \cap Y$, we have $N(S) \supseteq T$; thus either $N(S) = T$ or $N(S) \supset T$.

(a) If $N(S) = T$ then, since the vertices in $S\backslash\{u\}$ are matched with the vertices in T, $|N(S)| = |S| - 1$, indicating that G has no matching saturating all vertices in X.
(b) If $N(S) \supset T$, there is a vertex y in $Y\backslash T$ adjacent to a vertex x in S. Since all vertices of H except u are matched under M, either $x = u$ or else x is matched with a vertex of H. Therefore $xy \notin M$. If y is M-saturated, with $yz \in M$, we grow H by adding the vertices y and z and the edges xy and yz. We are then back in case (i). If y is M-unsaturated, we grow H by adding the vertex y and the edge xy, resulting in case (ii). The (u, y)-path of H is then an M-augmenting path with origin u, as required.

Figure 5.13 illustrates the above tree-growing procedure.
The algorithm described above is known as the *Hungarian method*, and

FEUE WHEP

5662

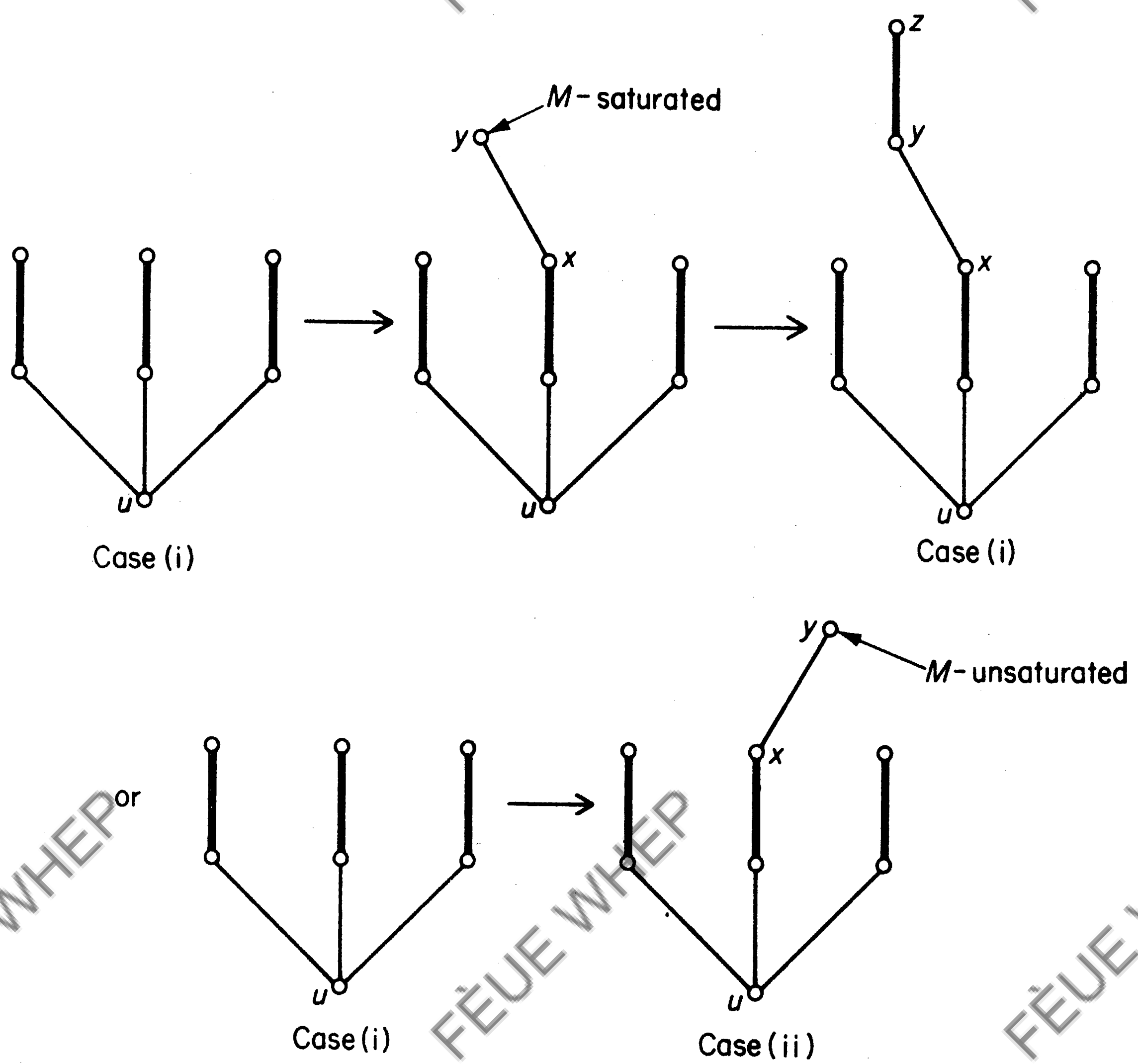


Figure 5.13. The tree-growing procedure

can be summarised as follows:

Start with an arbitrary matching M.

1. If M saturates every vertex in X, stop. Otherwise, let u be an M-unsaturated vertex in X. Set $S=\{u\}$ and $T=\emptyset$.
2. If $N(S)=T$ then $|N(S)|<|S|$, since $|T|=|S|-1$. Stop, since by Hall's theorem there is no matching that saturates every vertex in X. Otherwise, let $y\in N(S)\backslash T$.
3. If y is M-saturated, let $yz\in M$. Replace S by $S\cup\{z\}$ and T by $T\cup\{y\}$ and go to step 2. (Observe that $|T|=|S|-1$ is maintained after this replacement.) Otherwise, let P be an M-augmenting (u, y)-path. Replace M by $\hat{M}=M\,\Delta E(P)$ and go to step 1.

Consider, for example, the graph G in figure 5.14*a*, with initial matching $M=\{x_2y_2, x_3y_3, x_5y_5\}$. In figure 5.14*b* an M-alternating tree is grown, starting with x_1, and the M-augmenting path $x_1y_2x_2y_1$ found. This results in a new matching $\hat{M}=\{x_1y_2, x_2y_1, x_3y_3, x_5y_5\}$, and an $\hat{M}$-alternating tree is now grown from x_4 (figures 5.14*c* and 5.14*d*) Since there is no $\hat{M}$-augmenting

5663

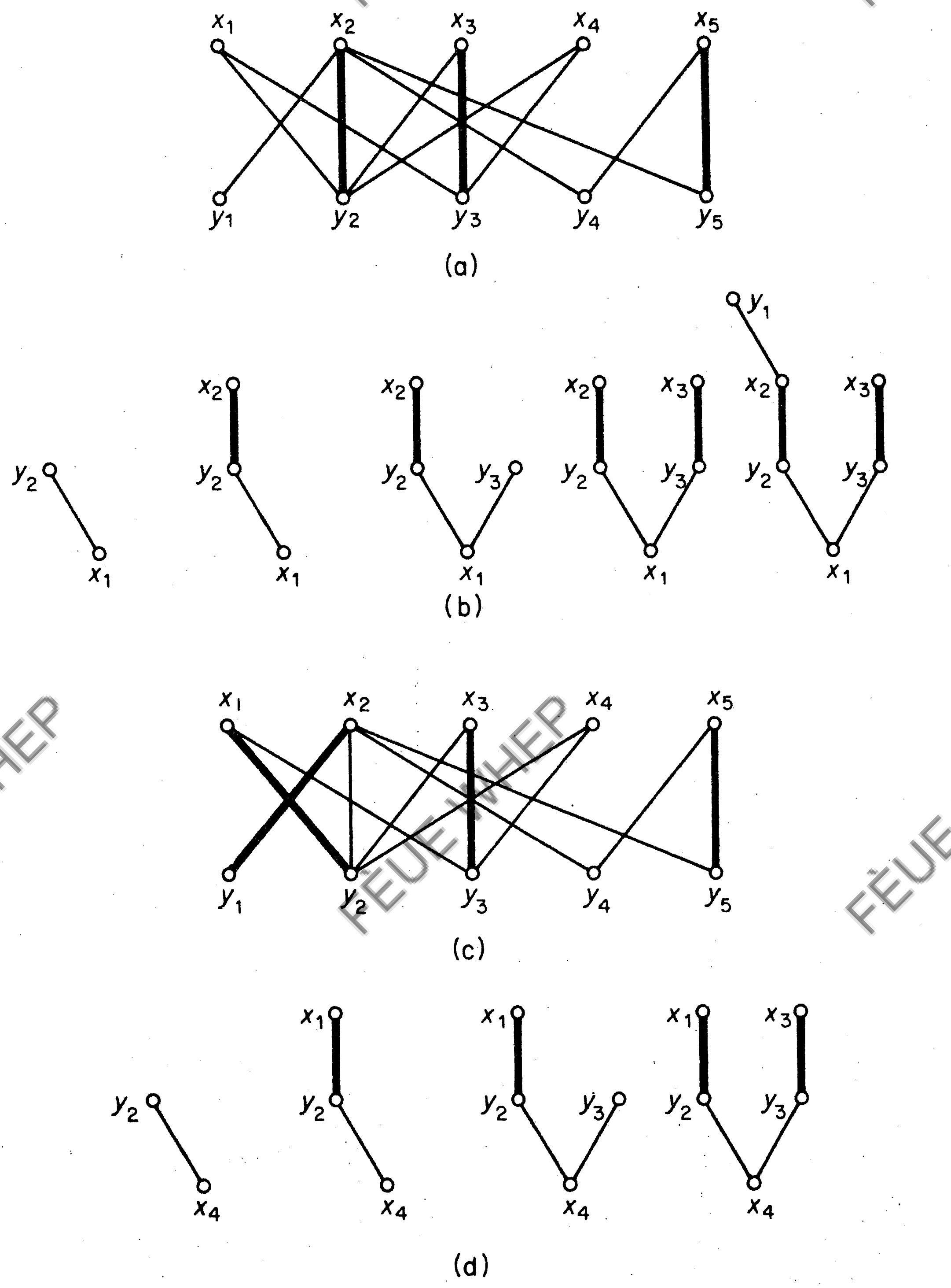


Figure 5.14. (*a*) Matching M; (*b*) an M-alternating tree; (*c*) matching $\hat{M}$; (*d*) an $\hat{M}$-alternating tree

path with origin x_4, the algorithm terminates. The set $S=\{x_1, x_3, x_4\}$, with neighbour set $N(S)=\{y_2, y_3\}$, shows that G has no perfect matching.

A flow diagram of the Hungarian method is given in figure 5.15. Since the algorithm can cycle through the tree-growing procedure, I, at most $|X|$ times before finding either an $S\subseteq X$ such that $|N(S)|<|S|$ or an M-augmenting path, and since the initial matching can be augmented at most $|X|$ times

FEUE WHEP

5664

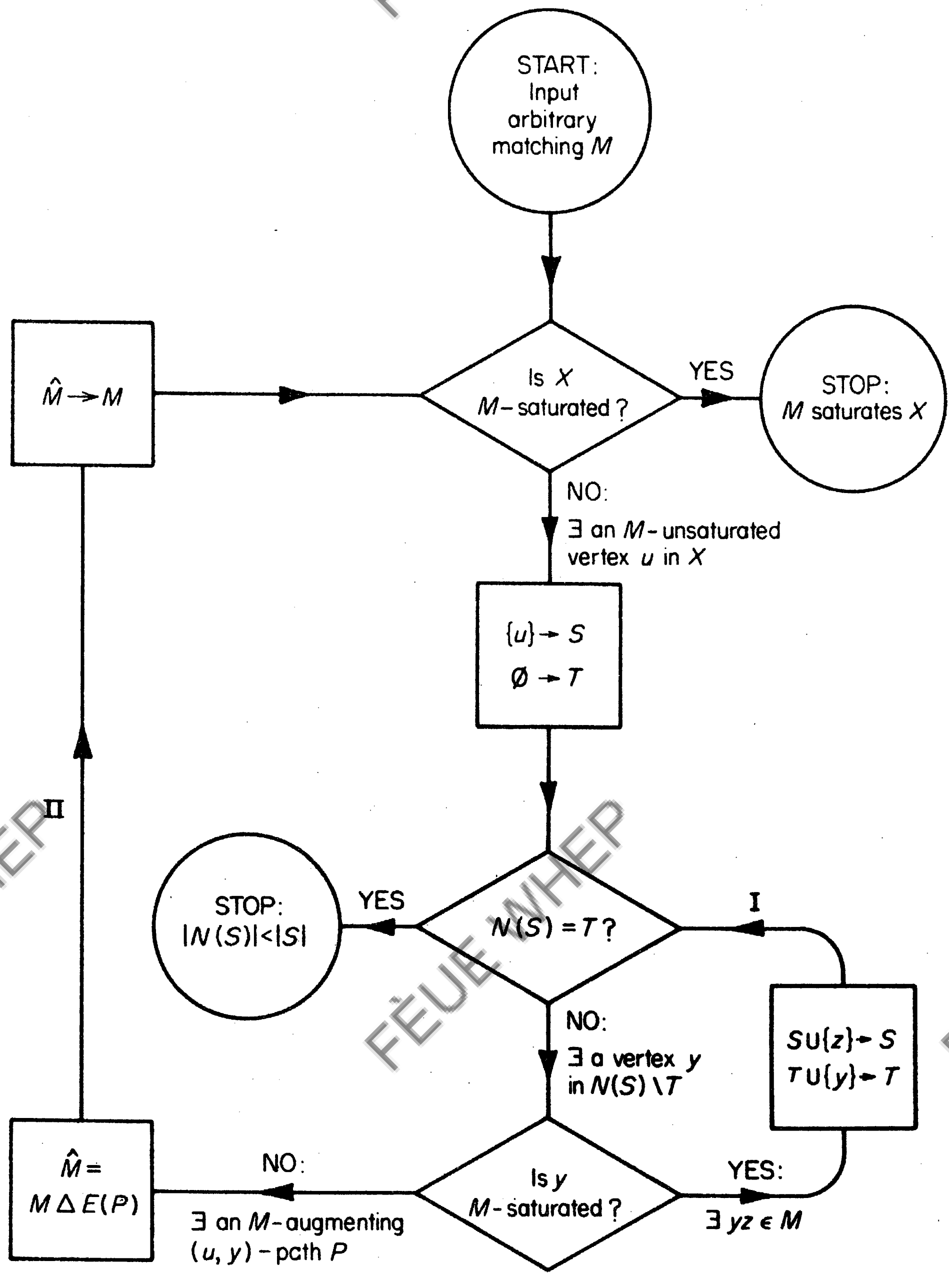


Figure 5.15. The Hungarian method

before a matching of the required type is found, it is clear that the Hungarian method is a good algorithm.

One can find a maximum matching in a bipartite graph by slightly modifying the above procedure (exercise 5.4.1). A good algorithm that determines such a matching in any graph has been given by Edmonds (1965).

Exercise

5.4.1 Describe how the Hungarian method can be used to find a maximum matching in a bipartite graph.

FEUE WHEP

5665

5.5 THE OPTIMAL ASSIGNMENT PROBLEM

The Hungarian method, described in section 5.4, is an efficient way of determining a feasible assignment of workers to jobs, if one exists. However one may, in addition, wish to take into account the effectiveness of the workers in their various jobs (measured, perhaps, by the profit to the company). In this case, one is interested in an assignment that maximises the total effectiveness of the workers. The problem of finding such an assignment is known as the *optimal assignment problem.*

Consider a weighted complete bipartite graph with bipartition (X, Y), where $X=\{x_1, x_2, \ldots, x_n\}$, $Y=\{y_1, y_2, \ldots, y_n\}$ and edge x_iy_j has weight $w_{ij}=w(x_iy_j)$, the effectiveness of worker X_i in job Y_j. The optimal assignment problem is clearly equivalent to that of finding a maximum-weight perfect matching in this weighted graph. We shall refer to such a matching as an *optimal matching.*

To solve the optimal assignment problem it is, of course, possible to enumerate all $n!$ perfect matchings and find an optimal one among them. However, for large n, such a procedure would clearly be most inefficient. In this section we shall present a good algorithm for finding an optimal matching in a weighted complete bipartite graph.

We define a *feasible vertex labelling* as a real-valued function l on the vertex set $X \cup Y$ such that, for all $x \in X$ and $y \in Y$

$$l(x)+l(y) \geq w(xy) \tag{5.11}$$

(The real number $l(v)$ is called the *label* of the vertex v.) A feasible vertex labelling is thus a labelling of the vertices such that the sum of the labels of the two ends of an edge is at least as large as the weight of the edge. No matter what the edge weights are, there always exists a feasible vertex labelling; one such is the function l given by

$$\left.\begin{array}{lll} l(x)=\max\limits_{y\in Y} w(xy) & \text{if} & x \in X \\ l(y)=0 & \text{if} & y \in Y \end{array}\right\} \tag{5.12}$$

If l is a feasible vertex labelling, we denote by E_l the set of those edges for which equality holds in (5.11); that is

$$E_l=\{xy \in E \mid l(x)+l(y)=w(xy)\}$$

The spanning subgraph of G with edge set E_l is referred to as the *equality subgraph* corresponding to the feasible vertex labelling l, and is denoted by G_l. The connection between equality subgraphs and optimal matchings is provided by the following theorem.

Theorem 5.5 Let l be a feasible vertex labelling of G. If G_l contains a perfect matching M^*, then M^* is an optimal matching of G.

5666

Proof Suppose that G_l contains a perfect matching M^*. Since G_l is a spanning subgraph of G, M^* is also a perfect matching of G. Now

$$w(M^*) = \sum_{e \in M^*} w(e) = \sum_{v \in V} l(v) \tag{5.13}$$

since each $e \in M^*$ belongs to the equality subgraph and the ends of edges of M^* cover each vertex exactly once. On the other hand, if M is any perfect matching of G, then

$$w(M) = \sum_{e \in M} w(e) \leq \sum_{v \in V} l(v) \tag{5.14}$$

It follows from (5.13) and (5.14) that $w(M^*) \geq w(M)$. Thus M^* is an optimal matching □

The above theorem is the basis of an algorithm, due to Kuhn (1955) and Munkres (1957), for finding an optimal matching in a weighted complete bipartite graph. Our treatment closely follows Edmonds (1967).

Starting with an arbitrary feasible vertex labelling l (for example, the one given in (5.12)), we determine G_l, choose an arbitrary matching M in G_l and apply the Hungarian method. If a perfect matching is found in G_l then, by theorem 5.5, this matching is optimal. Otherwise, the Hungarian method terminates in a matching M' that is not perfect, and an M'-alternating tree H that contains no M'-augmenting path and cannot be grown further (in G_l). We then modify l to a feasible vertex labelling $\hat{l}$ with the property that both M' and H are contained in $G_{\hat{l}}$ and H can be extended in $G_{\hat{l}}$. Such modifications in the feasible vertex labelling are made whenever necessary, until a perfect matching is found in some equality subgraph.

The Kuhn–Munkres Algorithm

Start with an arbitrary feasible vertex labelling l, determine G_l, and choose an arbitrary matching M in G_l.

1. If X is M-saturated, then M is a perfect matching (since $|X| = |Y|$) and hence, by theorem 5.5, an optimal matching; in this case, stop. Otherwise, let u be an M-unsaturated vertex. Set $S = \{u\}$ and $T = \emptyset$.
2. If $N_{G_l}(S) \supset T$, go to step 3. Otherwise, $N_{G_l}(S) = T$. Compute

$$\alpha_l = \min_{\substack{x \in S \\ y \notin T}} \{l(x) + l(y) - w(xy)\}$$

and the feasible vertex labelling $\hat{l}$ given by

$$\hat{l}(v) = \begin{cases} l(v) - \alpha_l & \text{if} \quad v \in S \\ l(v) + \alpha_l & \text{if} \quad v \in T \\ l(v) & \text{otherwise} \end{cases}$$

(Note that $\alpha_l > 0$ and that $N_{G_{\hat{l}}}(S) \supset T$.) Replace l by $\hat{l}$ and G_l by $G_{\hat{l}}$.

FEUE WHEP

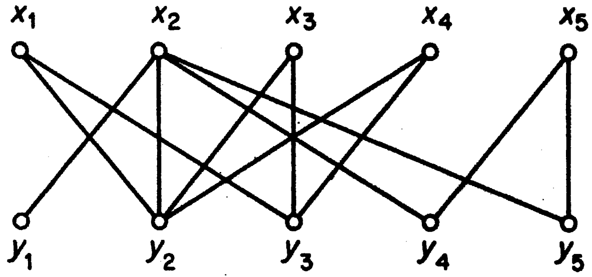
5667

$$\begin{bmatrix} 3 & 5 & 5 & 4 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 2 & 4 & 4 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 3 & 3 \end{bmatrix}$$

(a)

$$\begin{array}{c} \begin{bmatrix} 3 & \mathbf{5} & \mathbf{5} & 4 & 1 \\ \mathbf{2} & \mathbf{2} & 0 & \mathbf{2} & \mathbf{2} \\ 2 & \mathbf{4} & \mathbf{4} & 1 & 0 \\ 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 1 & 2 & 1 & \mathbf{3} & \mathbf{3} \end{bmatrix} \begin{matrix} 5 \\ 2 \\ 4 \\ 1 \\ 3 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 & 0 & 0 \end{matrix} \end{array}$$

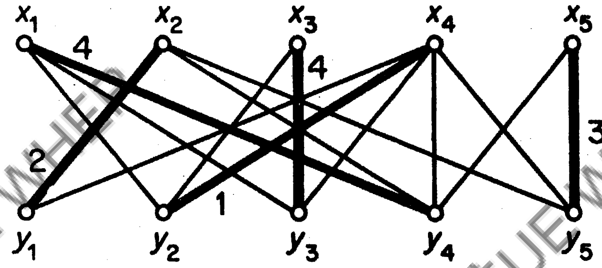
(b)



(c)

$$\begin{array}{c} \begin{bmatrix} 3 & \mathbf{5} & \mathbf{5} & \mathbf{4} & 1 \\ \mathbf{2} & 2 & 0 & \mathbf{2} & \mathbf{2} \\ 2 & \mathbf{4} & \mathbf{4} & 1 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 1 & 2 & 1 & \mathbf{3} & \mathbf{3} \end{bmatrix} \begin{matrix} 4 \\ 2 \\ 3 \\ 0 \\ 3 \end{matrix} \\ \begin{matrix} 0 & 1 & 1 & 0 & 0 \end{matrix} \end{array}$$

(d)



(e)

Figure 5.16

3. Choose a vertex y in $N_{G_l}(S)\backslash T$. As in the tree-growing procedure of section 5.4, consider whether or not y is M-saturated. If y is M-saturated, with $yz \in M$, replace S by $S \cup \{z\}$ and T by $T \cup \{y\}$, and go to step 2. Otherwise, let P be an M-augmenting (u, y)-path in G_l, replace M by $\hat{M} = M \,\Delta E(P)$, and go to step 1.

In illustrating the Kuhn–Munkres algorithm, it is convenient to represent a weighted complete bipartite graph G by a matrix $\mathbf{W} = [w_{ij}]$, where w_{ij} is the weight of edge x_iy_j in G. We shall start with the matrix of figure 5.16*a*. In figure 5.16*b*, the feasible vertex labelling (5.12) is shown (by placing the label of x_i to the right of row i of the matrix and the label of y_j below column j) and the entries corresponding to edges of the associated equality subgraph are indicated; the equality subgraph itself is depicted (without weights) in figure 5.16*c*. It was shown in the previous section that this graph has no perfect matching (the set $S = \{x_1, x_3, x_4\}$ has neighbour set $\{y_2, y_3\}$). We therefore modify our initial feasible vertex labelling to the one given in figure 5.16*d*. An application of the Hungarian method now shows that the associated equality subgraph (figure 5.16*e*) has the perfect matching $\{x_1y_4, x_2y_1, x_3y_3, x_4y_2, x_5y_5\}$. This is therefore an optimal matching of G.

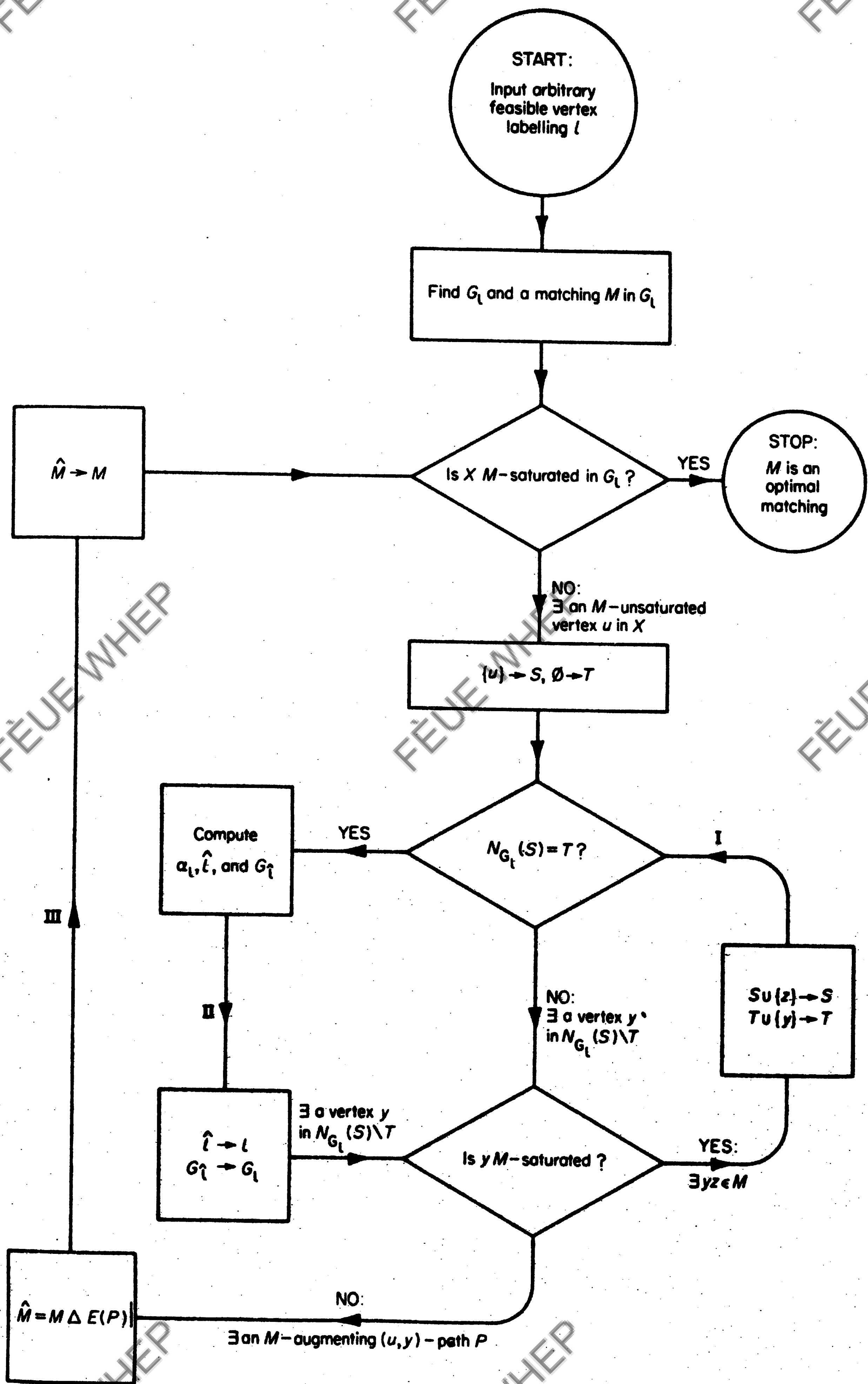


Figure 5.17. The Kuhn–Munkres algorithm

FEUE WHEP

5669

A flow diagram for the Kuhn–Munkres algorithm is given in figure 5.17. In cycle II, the number of computations required to compute $G_{\hat{l}}$ is clearly of order ν^2. Since the algorithm can cycle through I and II at most $|X|$ times before finding an M-augmenting path, and since the initial matching can be augmented at most $|X|$ times before an optimal matching is found, we see that the Kuhn–Munkres algorithm is a good algorithm.

Exercise

5.5.1 A *diagonal* of an $n \times n$ matrix is a set of n entries no two of which belong to the same row or the same column. The weight of a diagonal is the sum of the entries in it. Find a minimum-weight diagonal in the following matrix:

$$\begin{bmatrix} 4 & 5 & 8 & 10 & 11 \\ 7 & 6 & 5 & 7 & 4 \\ 8 & 5 & 12 & 9 & 6 \\ 6 & 6 & 13 & 10 & 7 \\ 4 & 5 & 7 & 9 & 8 \end{bmatrix}$$

REFERENCES

Anderson, I. (1971). Perfect matchings of a graph. *J. Combinatorial Theory B*, **10,** 183–86

Berge, C. (1957). Two theorems in graph theory. *Proc. Nat. Acad. Sci. USA*, **43,** 842–44

Edmonds, J. (1965). Paths, trees and flowers. *Canad. J. Math.*, **17,** 449–67

Edmonds, J. (1967). 'An Introduction to Matchings', notes of lectures delivered at the University of Michigan (unpublished)

Hall, P. (1935). On representatives of subsets. *J. London Math. Soc.*, **10,** 26–30

König, D. (1931). Graphs and matrices (Hungarian). *Mat. Fiz. Lapok*, **38,** 116–19.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, **2,** 83–97

Lovász, L. (1975). Three short proofs in graph theory. *J. Combinatorial Theory*, *B*, **19,** 111–13.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.*, **5,** 32–38

Petersen, J. (1891). Die Theorie der regulären graphs, *Acta Math.*, **15,** 193–220

Tutte, W. T. (1947). The factorization of linear graphs. *J. London Math. Soc.*, **22,** 107–11

5670

FEUE WHEP

6 Edge Colourings

6.1 EDGE CHROMATIC NUMBER

A *k-edge colouring* $\mathscr{C}$ of a loopless graph G is an assignment of k colours, $1, 2, \ldots, k$, to the edges of G. The colouring $\mathscr{C}$ is *proper* if no two adjacent edges have the same colour.

Alternatively, a k-edge colouring can be thought of as a partition $(E_1, E_2, \ldots, E_k)$ of E, where E denotes the (possibly empty) subset of E assigned colour i. A proper k-edge colouring is then a k-edge colouring $(E_1, E_2, \ldots, E_k)$ in which each subset E_i is a matching. The graph of figure 6.1 has the proper 4-edge colouring $(\{a, g\}, \{b, e\}, \{c, f\}, \{d\})$.

G is *k-edge colourable* if G has a proper k-edge-colouring. Trivially, every loopless graph G is ε-edge-colourable; and if G is k-edge-colourable, then G is also l-edge-colourable for every $l > k$. The *edge chromatic number* $\chi'(G)$, of a loopless graph G, is the minimum k for which G is k-edge-colourable. G is *k-edge-chromatic* if $\chi'(G) = k$. It can be readily verified that the graph of figure 6.1 has no proper 3-edge colouring. This graph is therefore 4-edge-chromatic.

Clearly, in any proper edge colouring, the edges incident with any one vertex must be assigned different colours. It follows that

$$\chi' \geq \Delta \tag{6.1}$$

Referring to the example of figure 6.1, we see that inequality (6.1) may be strict. However, we shall show that, in the case when G is bipartite, $\chi' = \Delta$. The following simple lemma is basic to our proof. We say that colour i is *represented* at vertex v if some edge incident with v has colour i.

Lemma 6.1.1 Let G be a connected graph that is not an odd cycle. Then

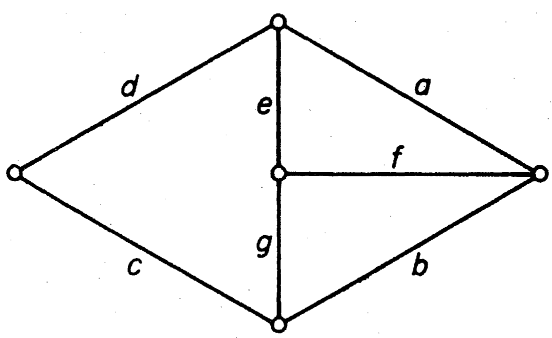


Figure 6.1

5671

G has a 2-edge colouring in which both colours are represented at each vertex of degree at least two.

Proof We may clearly assume that G is nontrivial. Suppose, first, that G is eulerian. If G is an even cycle, the proper 2-edge colouring of G has the required property. Otherwise, G has a vertex v_0 of degree at least four. Let $v_0e_1v_1 \ldots e_\varepsilon v_0$ be an Euler tour of G, and set

$$E_1 = \{e_i \mid i \text{ odd}\} \quad \text{and} \quad E_2 = \{e_i \mid i \text{ even}\} \tag{6.2}$$

Then the 2-edge colouring (E_1, E_2) of G has the required property, since each vertex of G is an internal vertex of $v_0e_1v_1 \ldots e_\varepsilon v_0$.

If G is not eulerian, construct a new graph G^* by adding a new vertex v_0 and joining it to each vertex of odd degree in G. Clearly G^* is eulerian. Let $v_0e_1v_1 \ldots e_{\varepsilon^*} v_0$ be an Euler tour of G^* and define E_1 and E_2 as in (6.2). It is then easily verified that the 2-edge colouring $(E_1 \cap E, E_2 \cap E)$ of G has the required property $\square$

Given a k-edge colouring $\mathscr{C}$ of G we shall denote by $c(v)$ the number of distinct colours represented at v. Clearly, we always have

$$c(v) \leq d(v) \tag{6.3}$$

Moreover, $\mathscr{C}$ is a proper k-edge colouring if and only if equality holds in (6.3) for all vertices v of G. We shall call a k-edge colouring $\mathscr{C}'$ an *improvement* on $\mathscr{C}$ if

$$\sum_{v \in V} c'(v) > \sum_{v \in V} c(v)$$

where $c'(v)$ is the number of distinct colours represented at v in the colouring $\mathscr{C}'$. An *optimal* k-edge colouring is one which cannot be improved.

Lemma 6.1.2 Let $\mathscr{C} = (E_1, E_2, \ldots, E_k)$ be an optimal k-edge colouring of G. If there is a vertex u in G and colours i and j such that i is not represented at u and j is represented at least twice at u, then the component of $G[E_i \cup E_j]$ that contains u is an odd cycle.

Proof Let u be a vertex that satisfies the hypothesis of the lemma, and denote by H the component of $G[E_i \cup E_j]$ containing u. Suppose that H is not an odd cycle. Then, by lemma 6.1.1, H has a 2-edge colouring in which both colours are represented at each vertex of degree at least two in H. When we recolour the edges of H with colours i and j in this way, we obtain a new k-edge colouring $\mathscr{C}' = (E'_1, E'_2, \ldots, E'_k)$ of G. Denoting by $c'(v)$ the number of distinct colours at v in the colouring $\mathscr{C}'$, we have

$$c'(u) = c(u) + 1$$

5672

since, now, both i and j are represented at u, and also

$$c'(v) \geq c(v) \quad \text{for} \quad v \neq u$$

Thus $\sum_{v \in V} c'(v) > \sum_{v \in V} c(v)$, contradicting the choice of $\mathscr{C}$. It follows that H is indeed an odd cycle □

Theorem 6.1 If G is bipartite, then $\chi' = \Delta$.

Proof Let G be a graph with $\chi' > \Delta$, let $\mathscr{C} = (E_1, E_2, \ldots, E_\Delta)$ be an optimal Δ-edge colouring of G, and let u be a vertex such that $c(u) < d(u)$. Clearly, u satisfies the hypothesis of lemma 6.1.2. Therefore G contains an odd cycle and so is not bipartite. It follows from (6.1) that if G is bipartite, then $\chi' = \Delta$ □

An alternative proof of theorem 6.1, using exercise 5.2.3a, is outlined in exercise 6.1.3.

Exercises

6.1.1 Show, by finding an appropriate edge colouring, that $\chi'(K_{m,n}) = \Delta(K_{m,n})$.

6.1.2 Show that the Petersen graph is 4-edge-chromatic.

6.1.3 (a) Show that if G is bipartite, then G has a Δ-regular bipartite supergraph.
(b) Using (a) and exercise 5.2.3a, give an alternative proof of theorem 6.1.

6.1.4 Describe a good algorithm for finding a proper Δ-edge colouring of a bipartite graph G.

6.1.5 Using exercise 1.5.8 and theorem 6.1, show that if G is loopless with $\Delta = 3$, then $\chi' \leq 4$.

6.1.6 Show that if G is bipartite with $\delta > 0$, then G has a δ-edge colouring such that all δ colours are represented at each vertex.
(R. P. Gupta)

6.2 VIZING'S THEOREM

As has already been noted, if G is not bipartite then we cannot necessarily conclude that $\chi' = \Delta$. An important theorem due to Vizing (1964) and, independently, Gupta (1966), asserts that, for any simple graph G, either $\chi' = \Delta$ or $\chi' = \Delta + 1$. The proof given here is by Fournier (1973).

Theorem 6.2 If G is simple, then either $\chi' = \Delta$ or $\chi' = \Delta + 1$.

Proof Let G be a simple graph. By virtue of (6.1) we need only show that $\chi' \leq \Delta + 1$. Suppose, then, that $\chi' > \Delta + 1$. Let $\mathscr{C} = (E_1, E_2, \ldots, E_{\Delta+1})$ be

5673

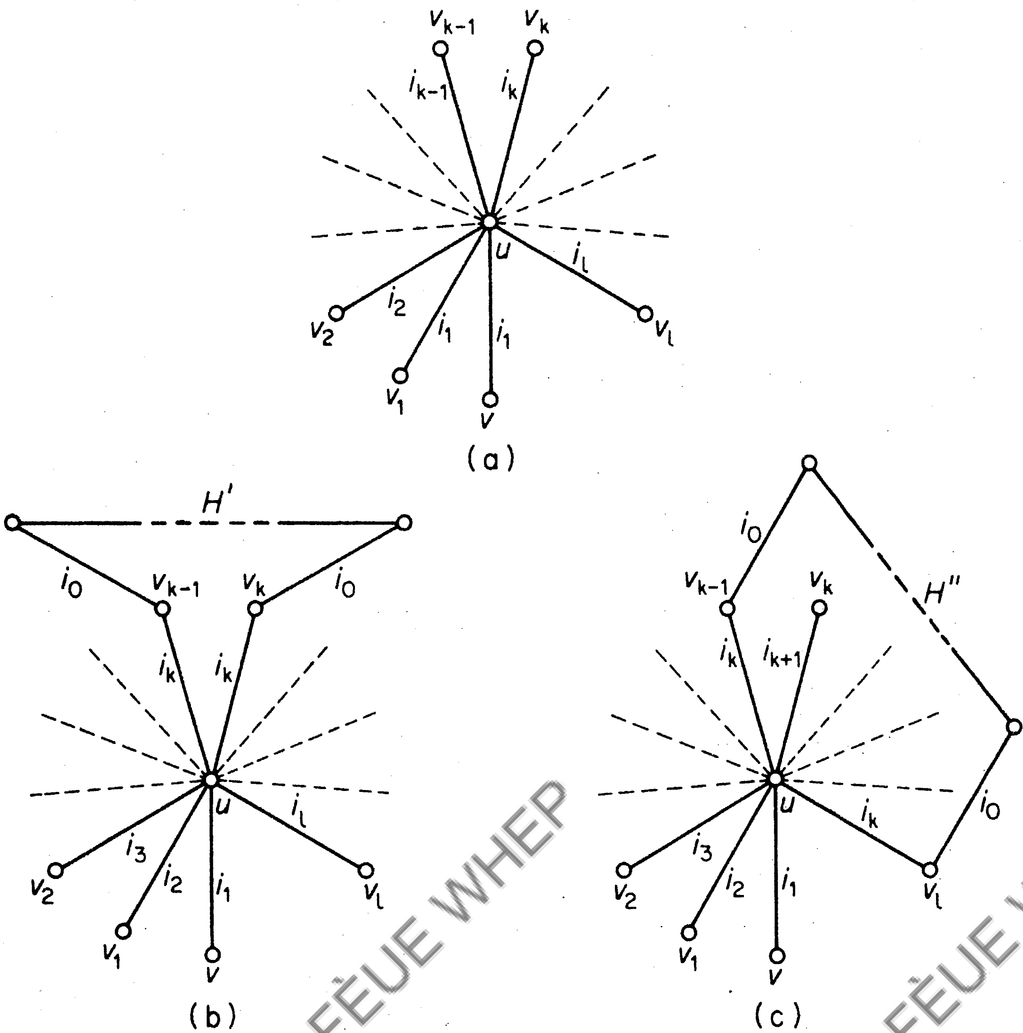


Figure 6.2

an optimal $(\Delta+1)$-edge colouring of G and let u be a vertex such that $c(u) < d(u)$. Then there exist colours i_0 and i_1 such that i_0 is not represented at u, and i_1 is represented at least twice at u. Let uv_1 have colour i_1, as in figure 6.2*a*.

Since $d(v_1) < \Delta+1$, some colour i_2 is not represented at v_1. Now i_2 must be represented at u since otherwise, by recolouring uv_1 with i_2, we would obtain an improvement on $\mathscr{C}$. Thus some edge uv_2 has colour i_2. Again, since $d(v_2) < \Delta+1$, some colour i_3 is not represented at v_2; and i_3 must be represented at u since otherwise, by recolouring uv_1 with i_2 and uv_2 with i_3, we would obtain an improved $(\Delta+1)$-edge colouring. Thus some edge uv_3 has colour i_3. Continuing this procedure we construct a sequence $v_1, v_2, \ldots$ of vertices and a sequence $i_1, i_2, \ldots$ of colours, such that

(i) uv_j has colour i_j, and
(ii) i_{j+1} is not represented at v_j.

Since the degree of u is finite, there exists a smallest integer l such that, for some $k < l$,

(iii) $i_{l+1} = i_k$.

FEUE WHEP

5674

The situation is depicted in figure 6.2*a*.

We now recolour G as follows. For $1 \leq j \leq k-1$, recolour uv_j with colour i_{j+1}, yielding a new $(\Delta+1)$-edge colouring $\mathscr{C}' = (E_1', E_2', \ldots, E_{\Delta+1}')$ (figure 6.2*b*). Clearly

$$c'(v) \geq c(v) \quad \text{for all} \quad v \in V$$

and therefore $\mathscr{C}'$ is also an optimal $(\Delta+1)$-edge colouring of G. By lemma 6.1.2, the component H' of $G[E_{i_0}' \cup E_{i_k}']$ that contains u is an odd cycle.

Now, in addition, recolour uv_j with colour i_{j+1}, $k \leq j \leq l-1$, and uv_l with colour i_k, to obtain a $(\Delta+1)$-edge colouring $\mathscr{C}'' = (E_1'', E_2'', \ldots, E_{\Delta+1}'')$ (figure 6.2*c*). As above

$$c''(v) \geq c(v) \quad \text{for all} \quad v \in V$$

and the component H'' of $G[E_{i_0}'' \cup E_{i_k}'']$ that contains u is an odd cycle. But, since v_k has degree two in H', v_k clearly has degree one in H''. This contradiction establishes the theorem □

Actually, Vizing proved a more general theorem than that given above, one that is valid for all loopless graphs. The maximum number of edges joining two vertices in G is called the *multiplicity* of G, and denoted by $\mu(G)$. We can now state Vizing's theorem in its full generality: if G is loopless, then $\Delta \leq \chi' \leq \Delta + \mu$.

This theorem is best possible in the sense that, for any μ, there exists a graph G such that $\chi' = \Delta + \mu$. For example, in the graph G of figure 6.3, $\Delta = 2\mu$ and, since any two edges are adjacent, $\chi' = \varepsilon = 3\mu$.

Strong as theorem 6.2 is, it leaves open one interesting question: which simple graphs satisfy $\chi' = \Delta$? The significance of this question will become apparent in chapter 9, when we study edge colourings of planar graphs.

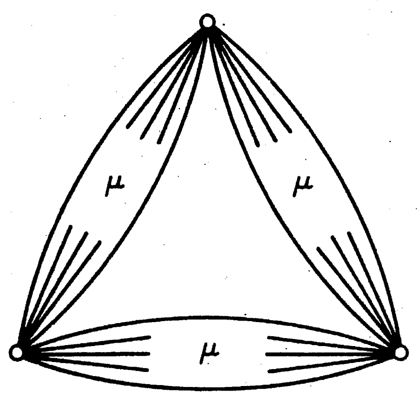


Figure 6.3. A graph G with $\chi' = \Delta + \mu$

5675

Exercises

6.2.1* Show, by finding appropriate edge colourings, that $\chi'(K_{2n-1}) = \chi'(K_{2n}) = 2n-1$.

6.2.2 Show that if G is a nonempty regular simple graph with ν odd, then $\chi' = \Delta + 1$.

6.2.3 (*a*) Let G be a simple graph. Show that if $\nu = 2n+1$ and $\varepsilon > n\Delta$, then $\chi' = \Delta + 1$. (V. G. Vizing)

(*b*) Using (*a*), show that

(i) if G is obtained from a simple regular graph with an even number of vertices by subdividing one edge, then $\chi' = \Delta + 1$;

(ii) if G is obtained from a simple k-regular graph with an odd number of vertices by deleting fewer than $k/2$ edges, then $\chi' = \Delta + 1$. (L. W. Beineke and R. J. Wilson)

6.2.4 (*a*) Show that if G is loopless, then G has a Δ-regular loopless supergraph.

(*b*) Using (*a*) and exercise 5.2.3*b*, show that if G is loopless and Δ is even, then $\chi' \leq 3\Delta/2$.

(Shannon, 1949 has shown that this inequality also holds when Δ is odd.)

6.2.5 G is called *uniquely k-edge-colourable* if any two proper k-edge colourings of G induce the same partition of E. Show that every uniquely 3-edge-colourable 3-regular graph is hamiltonian. (D. L. Greenwell and H. V. Kronk)

6.2.6 The *product* of simple graphs G and H is the simple graph $G \times H$ with vertex set $V(G) \times V(H)$, in which (u, v) is adjacent to (u', v') if and only if either $u = u'$ and $vv' \in E(H)$ or $v = v'$ and $uu' \in E(G)$.

(*a*) Using Vizing's theorem (6.2), show that $\chi'(G \times K_2) = \Delta(G \times K_2)$.

(*b*) Deduce that if H is nontrivial with $\chi'(H) = \Delta(H)$, then $\chi'(G \times H) = \Delta(G \times H)$.

6.2.7 Describe a good algorithm for finding a proper $(\Delta+1)$-edge colouring of a simple graph G.

6.2.8* Show that if G is simple with $\delta > 1$, then G has a $(\delta - 1)$-edge colouring such that all $\delta - 1$ colours are represented at each vertex. (R. P. Gupta)

APPLICATIONS

6.3 THE TIMETABLING PROBLEM

In a school, there are m teachers $X_1, X_2, \ldots, X_m$, and n classes $Y_1, Y_2, \ldots, Y_n$. Given that teacher X_i is required to teach class Y_j for p_{ij} periods, schedule a complete timetable in the minimum possible number of periods.

FEUE WHEP

5676

The above problem is known as the *timetabling problem*, and can be solved completely using the theory of edge colourings developed in this chapter. We represent the teaching requirements by a bipartite graph G with bipartition (X, Y), where $X = \{x_1, x_2, \ldots, x_m\}$, $Y = \{y_1, y_2, \ldots, y_n\}$ and vertices x_i and y_j are joined by p_{ij} edges. Now, in any one period, each teacher can teach at most one class, and each class can be taught by at most one teacher—this, at least, is our assumption. Thus a teaching schedule for one period corresponds to a matching in the graph and, conversely, each matching corresponds to a possible assignment of teachers to classes for one period. Our problem, therefore, is to partition the edges of G into as few matchings as possible or, equivalently, to properly colour the edges of G with as few colours as possible. Since G is bipartite, we know, by theorem 6.1, that $\chi' = \Delta$. Hence, if no teacher teaches for more than p periods, and if no class is taught for more than p periods, the teaching requirements can be scheduled in a p-period timetable. Furthermore, there is a good algorithm for constructing such a timetable, as is indicated in exercise 6.1.4. We thus have a complete solution to the timetabling problem.

However, the situation might not be so straightforward. Let us assume that only a limited number of classrooms are available. With this additional constraint, how many periods are now needed to schedule a complete timetable?

Suppose that altogether there are l lessons to be given, and that they have been scheduled in a p-period timetable. Since this timetable requires an average of l/p lessons to be given per period, it is clear that at least $\{l/p\}$ rooms will be needed in some one period. It turns out that one can always arrange l lessons in a p-period timetable so that at most $\{l/p\}$ rooms are occupied in any one period. This follows from theorem 6.3 below. We first have a lemma.

Lemma 6.3 Let M and N be disjoint matchings of G with $|M| > |N|$. Then there are disjoint matchings M' and N' of G such that $|M'| = |M| - 1$, $|N'| = |N| + 1$ and $M' \cup N' = M \cup N$.

Proof Consider the graph $H = G[M \cup N]$. As in the proof of theorem 5.1, each component of H is either an even cycle, with edges alternately in M and N, or else a path with edges alternately in M and N. Since $|M| > |N|$, some path component P of H must start and end with edges of M. Let $P = v_0 e_1 v_1 \ldots e_{2n+1} v_{2n+1}$, and set

$$M' = (M \backslash \{e_1, e_3, \ldots, e_{2n+1}\}) \cup \{e_2, e_4, \ldots, e_{2n}\}$$

$$N' = (N \backslash \{e_2, e_4, \ldots, e_{2n}\}) \cup \{e_1, e_3, \ldots, e_{2n+1}\}$$

Then M' and N' are matchings of G that satisfy the conditions of the lemma $\square$

FEUE WHEP

5677

$$\mathbf{P} = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{array} \begin{array}{c} \begin{array}{ccccc} Y_1 & Y_2 & Y_3 & Y_4 & Y_5 \end{array} \\ \left[\begin{array}{ccccc} 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] \end{array}$$

(a)

	Period 1	2	3	4
X_1	Y_1	Y_1	Y_3	Y_4
X_2	Y_2	–	Y_4	–
X_3	Y_3	Y_4	–	Y_2
X_4	Y_4	Y_5	–	–

(b)

Figure 6.4

Theorem 6.3 **If G is bipartite, and if $p \geq \Delta$, then there exist p disjoint matchings $M_1, M_2, \ldots, M_p$ of G such that**

$$E = M_1 \cup M_2 \cup \ldots \cup M_p \tag{6.4}$$

and, for $1 \leq i \leq p$

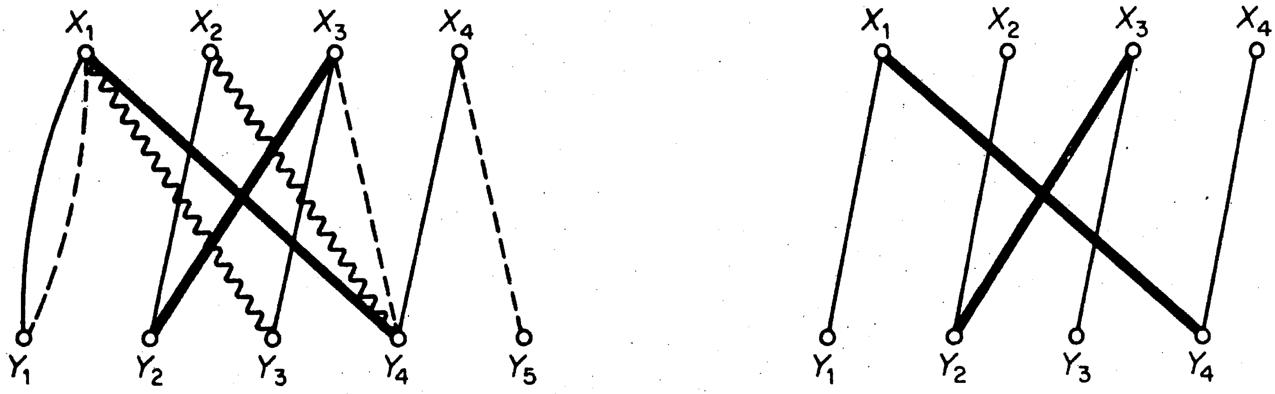
$$[\varepsilon/p] \leq |M_i| \leq \{\varepsilon/p\} \tag{6.5}$$

(Note: condition (6.5) says that any two matchings M_i and M_j differ in size by at most one.)

Proof Let G be a bipartite graph. By theorem 6.1, the edges of G can be partitioned into Δ matchings $M'_1, M'_2, \ldots, M'_\Delta$. Therefore, for any $p \geq \Delta$, there exist p disjoint matchings $M'_1, M'_2, \ldots, M'_p$ (with $M'_i = \emptyset$ for $i > \Delta$) such that

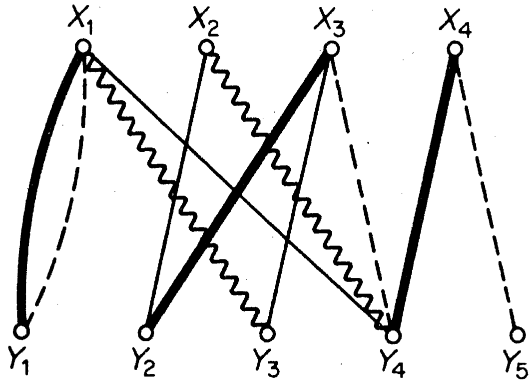
$$E = M'_1 \cup M'_2 \cup \ldots \cup M'_p$$

By repeatedly applying lemma 6.3 to pairs of these matchings that differ in size by more than one, we eventually obtain p disjoint matchings $M_1, M_2, \ldots, M_p$ of G satisfying (6.4) and (6.5), as required □



(a) (b)

Figure 6.5



(a)

	Period 1	2	3	4
x_1	y_4	y_1	y_3	y_1
x_2	y_2	–	y_4	–
x_3	y_3	y_4	–	y_2
x_4	–	y_5	–	y_4

(b)

Figure 6.6

As an example, suppose that there are four teachers and five classes, and that the teaching requirement matrix $\mathbf{P}=[p_{ij}]$ is as given in figure 6.4*a*. One possible 4-period timetable is shown in figure 6.4*b*.

We can represent the above timetable by a decomposition into matchings of the edge set of the bipartite graph G corresponding to $\mathbf{P}$, as shown in figure 6.5*a*. (Normal edges correspond to period 1, broken edges to period 2, wavy edges to period 3, and heavy edges to period 4.)

From the timetable we see that four classes are taught in period 1, and so four rooms are needed. However $\varepsilon=11$ and so, by theorem 6.4, a 4-period timetable can be arranged so that in each period either $2(=[11/4])$ or $3(=\{11/4\})$ classes are taught. Let M_1 denote the normal matching and M_4 the heavy matching; notice that $|M_1|=4$ and $|M_4|=2$. We can now find a 4-period 3-room timetable by considering $G[M_1\cup M_4]$ (figure 6.5*b*). $G[M_1\cup M_4]$ has two components, each consisting of a path of length three. Both paths start and end with normal edges and so, by interchanging the matchings on one of the two paths, we shall reduce the normal matching to one of three edges, and at the same time increase the heavy matching to one of three edges. If we choose the path $y_1x_1y_4x_4$, making the edges y_1x_1 and y_4x_4 heavy and the edge x_1y_4 normal, we obtain the decomposition of E shown in figure 6.6*a*. This then gives the revised timetable shown in figure 6.6*b*; here, only three rooms are needed at any one time.

	Period 1	2	3	4	5	6
x_1	y_4	y_3	y_1	–	y_1	–
x_2	y_2	y_4	–	–	–	–
x_3	–	–	y_4	y_3	y_2	–
x_4	–	–	–	y_4	–	y_5

Figure 6.7

5679

However, suppose that there are just two rooms available. Theorem 6.4 tells us that there must be a 6-period timetable that satisfies our requirements (since $\{11/6\} = 2$). Such a timetable is given in figure 6.7.

In practice, most problems on timetabling are complicated by preassignments (that is, conditions specifying the periods during which certain teachers and classes must meet). This generalisation of the timetabling problem has been studied by Dempster (1971) and de Werra (1970).

Exercise

6.3.1 In a school there are seven teachers and twelve classes. The teaching requirements for a five-day week are given by the matrix

$$\mathbf{P} = \begin{array}{c|cccccccccccc} & Y_1 & Y_2 & Y_3 & Y_4 & Y_5 & Y_6 & Y_7 & Y_8 & Y_9 & Y_{10} & Y_{11} & Y_{12} \\ \hline X_1 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ X_2 & 1 & 3 & 6 & 0 & 4 & 2 & 5 & 1 & 3 & 3 & 0 & 4 \\ X_3 & 5 & 0 & 5 & 5 & 0 & 0 & 5 & 0 & 5 & 0 & 5 & 5 \\ X_4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 4 & 2 & 3 \\ X_5 & 3 & 5 & 2 & 2 & 0 & 3 & 1 & 4 & 4 & 3 & 2 & 5 \\ X_6 & 5 & 5 & 0 & 0 & 5 & 5 & 0 & 5 & 0 & 5 & 5 & 0 \\ X_7 & 0 & 3 & 4 & 3 & 4 & 3 & 4 & 3 & 4 & 3 & 3 & 0 \end{array}$$

where p_{ij} is the number of periods that teacher X_i must teach class Y_j.

(*a*) Into how many periods must a day be divided so that the requirements can be satisfied?

(*b*) If an eight-period/day timetable is drawn up, how many classrooms will be needed?

REFERENCES

Dempster, M. A. H. (1971). Two algorithms for the time-table problem, in *Combinatorial Mathematics and its Applications* (ed. D. J. A. Welsh), Academic Press, New York, pp. 63–85

Fournier, J.-C. (1973). Colorations des arêtes d'un graphe. *Cahiers du CERO*, **15,** 311–14.

Gupta, R. P. (1966). The chromatic index and the degree of a graph. *Notices Amer. Math. Soc.*, **13,** abstract 66T-429

Shannon, C. E. (1949). A theorem on coloring the lines of a network. *J. Math. Phys.*, **28,** 148–51

Vizing, V. G. (1964). On an estimate of the chromatic class of a *p*-graph (Russian). *Diskret. Analiz.*, **3,** 25–30

de Werra, D. (1970). On some combinatorial problems arising in scheduling. *INFOR*, **8,** 165–75

5680

7 Independent Sets and Cliques

7.1 INDEPENDENT SETS

A subset S of V is called an *independent set* of G if no two vertices of S are adjacent in G. An independent set is *maximum* if G has no independent set S' with $|S'|>|S|$. Examples of independent sets are shown in figure 7.1.

Recall that a subset K of V such that every edge of G has at least one end in K is called a covering of G. The two examples of independent sets given in figure 7.1 are both complements of coverings. It is not difficult to see that this is always the case.

Theorem 7.1 A set $S \subseteq V$ is an independent set of G if and only if $V\backslash S$ is a covering of G.

Proof By definition, S is an independent set of G if and only if no edge of G has both ends in S or, equivalently, if and only if each edge has at least one end in $V\backslash S$. But this is so if and only if $V\backslash S$ is a covering of G. □

The number of vertices in a maximum independent set of G is called the *independence number* of G and is denoted by $\alpha(G)$; similarly, the number of vertices in a minimum covering of G is the *covering number* of G and is denoted by $\beta(G)$.

Corollary 7.1 $\alpha+\beta=\nu$.

Proof Let S be a maximum independent set of G, and let K be a minimum covering of G. Then, by theorem 7.1, $V\backslash K$ is an independent set

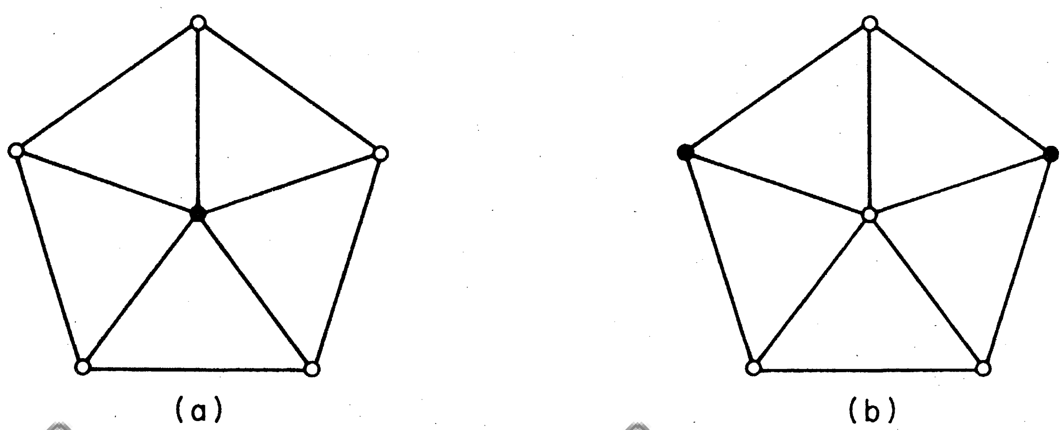


Figure 7.1. (*a*) An independent set; (*b*) a maximum independent set

FEUE WHEP

5681

FEUE WHEP

and $V \backslash S$ is a covering. Therefore

$$\nu-\beta=|V \backslash K| \leq \alpha \tag{7.1}$$

and

$$\nu-\alpha=|V \backslash S| \geq \beta \tag{7.2}$$

Combining (7.1) and (7.2) we have $\alpha+\beta=\nu$ □

The edge analogue of an independent set is a set of links no two of which are adjacent, that is, a matching. The edge analogue of a covering is called an edge covering. An *edge covering* of G is a subset L of E such that each vertex of G is an end of some edge in L. Note that edge coverings do not always exist; a graph G has an edge covering if and only if $\delta>0$. We denote the number of edges in a maximum matching of G by $\alpha'(G)$, and the number of edges in a minimum edge covering of G by $\beta'(G)$; the numbers $\alpha'(G)$ and $\beta'(G)$ are the *edge independence number* and *edge covering number* of G, respectively.

Matchings and edge coverings are not related to one another as simply as are independent sets and coverings; the complement of a matching need not be an edge covering, nor is the complement of an edge covering necessarily a matching. However, it so happens that the parameters α' and β' are related in precisely the same manner as are α and β.

Theorem 7.2 (Gallai, 1959) If $\delta>0$, then $\alpha'+\beta'=\nu$.

Proof Let M be a maximum matching in G and let U be the set of M-unsaturated vertices. Since $\delta>0$ and M is maximum, there exists a set E' of $|U|$ edges, one incident with each vertex in U. Clearly, $M \cup E'$ is an edge covering of G, and so

$$\beta' \leq|M \cup E'|=\alpha'+(\nu-2 \alpha')=\nu-\alpha'$$

or

$$\alpha'+\beta' \leq \nu \tag{7.3}$$

Now let L be a minimum edge covering of G, set $H=G[L]$ and let M be a maximum matching in H. Denote the set of M-unsaturated vertices in H by U. Since M is maximum, $H[U]$ has no links and therefore

$$|L|-|M|=|L \backslash M| \geq|U|=\nu-2|M|$$

Because H is a subgraph of G, M is a matching in G and so

$$\alpha'+\beta' \geq|M|+|L| \geq \nu \tag{7.4}$$

Combining (7.3) and (7.4), we have $\alpha'+\beta'=\nu$ □

We can now prove a theorem that bears a striking formal resemblance to König's theorem (5.3).

5682

Theorem 7.3 In a bipartite graph G with $\delta > 0$, the number of vertices in a maximum independent set is equal to the number of edges in a minimum edge covering.

Proof Let G be a bipartite graph with $\delta > 0$. By corollary 7.1 and theorem 7.2, we have

$$\alpha + \beta = \alpha' + \beta'$$

and, since G is bipartite, it follows from theorem 5.3 that $\alpha' = \beta$. Thus $\alpha = \beta'$ □

Even though the concept of an independent set is analogous to that of a matching, there exists no theory of independent sets comparable to the theory of matchings presented in chapter 5; for example, no good algorithm for finding a maximum independent set in a graph is known. However, there are two interesting theorems that relate the number of vertices in a maximum independent set of a graph to various other parameters of the graph. These theorems will be discussed in sections 7.2 and 7.3.

Exercises

7.1.1 (*a*) Show that G is bipartite if and only if $\alpha(H) \geq \frac{1}{2}\nu(H)$ for every subgraph H of G.
(*b*) Show that G is bipartite if and only if $\alpha(H) = \beta'(H)$ for every subgraph H of G such that $\delta(H) > 0$.

7.1.2 A graph is *α-critical* if $\alpha(G-e) > \alpha(G)$ for all $e \in E$. Show that a connected α-critical graph has no cut vertices.

7.1.3 A graph G is *β-critical* if $\beta(G-e) < \beta(G)$ for all $e \in E$. Show that
(*a*) a connected β-critical graph has no cut vertices;
(*b*)* if G is connected, then $\beta \leq \frac{1}{2}(\varepsilon + 1)$.

7.2 RAMSEY'S THEOREM

In this section we deal only with simple graphs. A *clique* of a simple graph G is a subset S of V such that $G[S]$ is complete. Clearly, S is a clique of G if and only if S is an independent set of G^c, and so the two concepts are complementary.

If G has no large cliques, then one might expect G to have a large independent set. That this is indeed the case was first proved by Ramsey (1930). He showed that, given any positive integers k and l, there exists a smallest integer $r(k, l)$ such that every graph on $r(k, l)$ vertices contains either a clique of k vertices or an independent set of l vertices. For example, it is easy to see that

$$r(1, l) = r(k, 1) = 1 \tag{7.5}$$

5683

and

$$r(2, l) = l, \qquad r(k, 2) = k \tag{7.6}$$

The numbers $r(k, l)$ are known as the *Ramsey numbers*. The following theorem on Ramsey numbers is due to Erdös and Szekeres (1935) and Greenwood and Gleason (1955).

Theorem 7.4 For any two integers $k \geq 2$ and $l \geq 2$

$$r(k, l) \leq r(k, l-1) + r(k-1, l) \tag{7.7}$$

Furthermore, if $r(k, l-1)$ and $r(k-1, l)$ are both even, then strict inequality holds in (7.7).

Proof Let G be a graph on $r(k, l-1) + r(k-1, l)$ vertices, and let $v \in V$. We distinguish two cases:

(i) v is nonadjacent to a set S of at least $r(k, l-1)$ vertices, or
(ii) v is adjacent to a set T of at least $r(k-1, l)$ vertices.

Note that either case (i) or case (ii) must hold because the number of vertices to which v is nonadjacent plus the number of vertices to which v is adjacent is equal to $r(k, l-1) + r(k-1, l) - 1$.

In case (i), $G[S]$ contains either a clique of k vertices or an independent set of $l-1$ vertices, and therefore $G[S \cup \{v\}]$ contains either a clique of k vertices or an independent set of l vertices. Similarly, in case (ii), $G[T \cup \{v\}]$ contains either a clique of k vertices or an independent set of l vertices. Since one of case (i) and case (ii) must hold, it follows that G contains either a clique of k vertices or an independent set of l vertices. This proves (7.7).

Now suppose that $r(k, l-1)$ and $r(k-1, l)$ are both even, and let G be a graph on $r(k, l-1) + r(k-1, l) - 1$ vertices. Since G has an odd number of vertices, it follows from corollary 1.1 that some vertex v is of even degree; in particular, v cannot be adjacent to precisely $r(k-1, l) - 1$ vertices. Consequently, either case (i) or case (ii) above holds, and therefore G contains either a clique of k vertices or an independent set of l vertices. Thus

$$r(k, l) \leq r(k, l-1) + r(k-1, l) - 1$$

as stated □

The determination of the Ramsey numbers in general is a very difficult unsolved problem. Lower bounds can be obtained by the construction of suitable graphs. Consider, for example, the four graphs in figure 7.2.

The 5-cycle (figure 7.2a) contains no clique of three vertices and no independent set of three vertices. It shows, therefore, that

$$r(3, 3) \geq 6 \tag{7.8}$$

5684

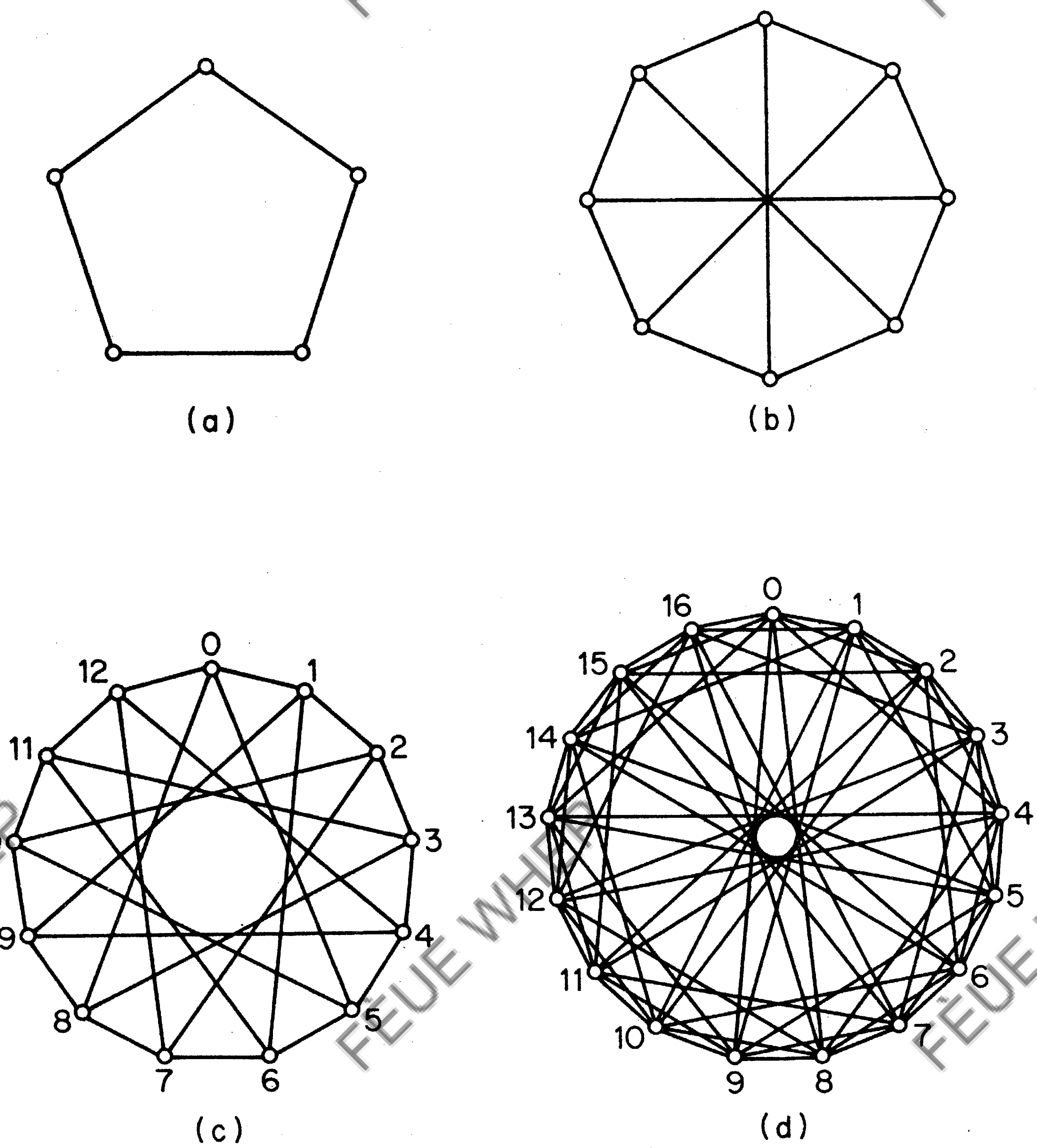


Figure 7.2. (*a*) A (3,3)-Ramsey graph; (*b*) a (3,4)-Ramsey graph; (*c*) a (3,5)-Ramsey graph; (*d*) a (4,4)-Ramsey graph

The graph of figure 7.2*b* contains no clique of three vertices and no independent set of four vertices. Hence

$$r(3,4) \geq 9 \tag{7.9}$$

Similarly, the graph of figure 7.2*c* shows that

$$r(3,5) \geq 14 \tag{7.10}$$

and the graph of figure 7.2*d* yields

$$r(4,4) \geq 18 \tag{7.11}$$

With the aid of theorem 7.4 and equations (7.6) we can now show that equality in fact holds in (7.8), (7.9), (7.10) and (7.11). Firstly, by (7.7) and (7.6)

$$r(3,3) \leq r(3,2) + r(2,3) = 6$$

FEUE WHEP

5685

and therefore, using (7.8), we have $r(3, 3)=6$. Noting that $r(3, 3)$ and $r(2, 4)$ are both even, we apply theorem 7.4 and (7.6) to obtain

$$r(3, 4)\leq r(3, 3)+r(2, 4)-1=9$$

With (7.9) this gives $r(3, 4)=9$. Now we again apply (7.7) and (7.6) to obtain

$$r(3, 5)\leq r(3, 4)+r(2, 5)=14$$

and

$$r(4, 4)\leq r(4, 3)+r(3, 4)=18$$

which, together with (7.10) and (7.11), respectively, yield $r(3, 5)=14$ and $r(4, 4)=18$.

The following table shows all Ramsey numbers $r(k, l)$ known to date.

$k \backslash l$	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	2	3	4	5	6	7
3	1	3	6	9	14	18	23
4	1	4	9	18			

A (k, l)-*Ramsey graph* is a graph on $r(k, l)-1$ vertices that contains neither a clique of k vertices nor an independent set of l vertices. By definition of $r(k, l)$ such graphs exist for all $k\geq 2$ and $l\geq 2$. Ramsey graphs often seem to possess interesting structures. All of the graphs in figure 7.2 are Ramsey graphs; the last two can be obtained from finite fields in the following way. We get the $(3, 5)$-Ramsey graph by regarding the thirteen vertices as elements of the field of integers modulo 13, and joining two vertices by an edge if their difference is a cubic residue of 13 (either 1, 5, 8 or 12); the $(4, 4)$-Ramsey graph is obtained by regarding the vertices as elements of the field of integers modulo 17, and joining two vertices if their difference is a quadratic residue of 17 (either 1, 2, 4, 8, 9, 13, 15 or 16). It has been conjectured that the (k, k)-Ramsey graphs are always self-complementary (that is, isomorphic to their complements); this is true for $k=2$, 3 and 4.

In general, theorem 7.4 yields the following upper bound for $r(k, l)$.

Theorem 7.5 $\quad r(k, l)\leq\binom{k+l-2}{k-1}$

Proof By induction on $k+l$. Using (7.5) and (7.6) we see that the theorem holds when $k+l\leq 5$. Let m and n be positive integers, and assume that the theorem is valid for all positive integers k and l such that

FEUE WHEP

5686

$5 \le k+l < m+n$. Then, by theorem 7.4 and the induction hypothesis

$$r(m, n) \le r(m, n-1) + r(m-1, n)$$
$$\le \binom{m+n-3}{m-1} + \binom{m+n-3}{m-2} = \binom{m+n-2}{m-1}$$

Thus the theorem holds for all values of k and l □

A lower bound for $r(k, k)$ is given in the next theorem. It is obtained by means of a powerful technique known as the *probabilistic method* (see Erdös and Spencer, 1974). The probabilistic method is essentially a crude counting argument. Although nonconstructive, it can often be applied to assert the existence of a graph with certain specified properties.

Theorem 7.6 (Erdös, 1947) $r(k, k) \ge 2^{k/2}$

Proof. Since $r(1, 1) = 1$ and $r(2, 2) = 2$, we may assume that $k \ge 3$. Denote by $\mathcal{G}_n$ the set of simple graphs with vertex set $\{v_1, v_2, \ldots, v_n\}$, and by $\mathcal{G}_n^k$ the set of those graphs in $\mathcal{G}_n$ that have a clique of k vertices. Clearly

$$|\mathcal{G}_n| = 2^{\binom{n}{2}} \tag{7.12}$$

since each subset of the $\binom{n}{2}$ possible edges v_iv_j determines a graph in $\mathcal{G}_n$. Similarly, the number of graphs in $\mathcal{G}_n$ having a particular set of k vertices as a clique is $2^{\binom{n}{2}-\binom{k}{2}}$. Since there are $\binom{n}{k}$ distinct k-element subsets of $\{v_1, v_2, \ldots, v_n\}$, we have

$$|\mathcal{G}_n^k| \le \binom{n}{k} 2^{\binom{n}{2}-\binom{k}{2}} \tag{7.13}$$

By (7.12) and (7.13)

$$\frac{|\mathcal{G}_n^k|}{|\mathcal{G}_n|} \le \binom{n}{k} 2^{-\binom{k}{2}} < \frac{n^k 2^{-\binom{k}{2}}}{k!} \tag{7.14}$$

Suppose, now, that $n < 2^{k/2}$. From (7.14) it follows that

$$\frac{|\mathcal{G}_n^k|}{|\mathcal{G}_n|} < \frac{2^{k^2/2} 2^{-\binom{k}{2}}}{k!} = \frac{2^{k/2}}{k!} < \tfrac{1}{2}$$

Therefore, fewer than half of the graphs in $\mathcal{G}_n$ contain a clique of k vertices. Also, because $\mathcal{G}_n = \{G \mid G^c \in \mathcal{G}_n\}$, fewer than half of the graphs in $\mathcal{G}_n$ contain an independent set of k vertices. Hence some graph in $\mathcal{G}_n$ contains neither a clique of k vertices nor an independent set of k vertices. Because this holds for any $n < 2^{k/2}$, we have $r(k, k) \ge 2^{k/2}$ □

From theorem 7.6 we can immediately deduce a lower bound for $r(k, l)$.

FEUE WHEP

5687

Corollary 7.6 If $m = \min\{k, l\}$, then $r(k, l) \geq 2^{m/2}$

All known lower bounds for $r(k, l)$ obtained by constructive arguments are much weaker than that given in corollary 7.6; the best is due to Abbott (1972), who shows that $r(2^n+1, 2^n+1) \geq 5^n+1$ (exercise 7.2.4).

The Ramsey numbers $r(k, l)$ are sometimes defined in a slightly different way from that given at the beginning of this section. One easily sees that $r(k, l)$ can be thought of as the smallest integer n such that every 2-edge colouring (E_1, E_2) of K_n contains either a complete subgraph on k vertices, all of whose edges are in colour 1, or a complete subgraph on l vertices, all of whose edges are in colour 2. Expressed in this form, the Ramsey numbers have a natural generalisation. We define $r(k_1, k_2, \ldots, k_m)$ to be the smallest integer n such that every m-edge colouring $(E_1, E_2, \ldots, E_m)$ of K_n contains, for some i, a complete subgraph on k_i vertices, all of whose edges are in colour i.

The following theorem and corollary generalise (7.7) and theorem 7.5, and can be proved in a similar manner. They are left as an exercise (7.2.2).

Theorem 7.7 $r(k_1, k_2, \ldots, k_m) \leq r(k_1-1, k_2, \ldots, k_m) + r(k_1, k_2-1, \ldots, k_m) + \ldots + r(k_1, k_2, \ldots, k_m-1) - m + 2$

Corollary 7.7 $r(k_1+1, k_2+1, \ldots, k_m+1) \leq \dfrac{(k_1+k_2+\ldots+k_m)!}{k_1!\,k_2!\ldots k_m!}$

Exercises

7.2.1 Show that, for all k and l, $r(k, l) = r(l, k)$.

7.2.2 Prove theorem 7.7 and corollary 7.7.

7.2.3 Let r_n denote the Ramsey number $r(k_1, k_2, \ldots, k_n)$ with $k_i = 3$ for all i.

(*a*) Show that $r_n \leq n(r_{n-1}-1)+2$.
(*b*) Noting that $r_2 = 6$, use (*a*) to show that $r_n \leq [n!\,e]+1$.
(*c*) Deduce that $r_3 \leq 17$.
(Greenwood and Gleason, 1955 have shown that $r_3 = 17$.)

7.2.4 The *composition* of simple graphs G and H is the simple graph $G[H]$ with vertex set $V(G)\times V(H)$, in which (u, v) is adjacent to (u', v') if and only if either $uu' \in E(G)$ or $u = u'$ and $vv' \in E(H)$.

(*a*) Show that $\alpha(G[H]) \leq \alpha(G)\alpha(H)$.
(*b*) Using (*a*), show that

$$r(kl+1, kl+1)-1 \geq (r(k+1, k+1)-1)\times(r(l+1, l+1)-1)$$

(*c*) Deduce that $r(2^n+1, 2^n+1) \geq 5^n+1$ for all $n \geq 0$.

(H. L. Abbott)

5688

7.2.5 Show that the join of a 3-cycle and a 5-cycle contains no K_6, but that every 2-edge colouring yields a monochromatic triangle.

(R. L. Graham)

(Folkman, 1970 has constructed a graph containing no K_4 in which every 2-edge colouring yields a monochromatic triangle—this graph has a very large number of vertices.)

7.2.6 Let $G_1, G_2, \ldots, G_m$ be simple graphs. The *generalised Ramsey number* $r(G_1, G_2, \ldots, G_m)$ is the smallest integer n such that every m-edge colouring $(E_1, E_2, \ldots, E_m)$ of K_n contains, for some i, a subgraph isomorphic to G_i in colour i. Show that

(*a*) if G is a path of length three and H is a 4-cycle, then $r(G, G) = 5$, $r(G, H) = 5$ and $r(H, H) = 6$;

(*b*)* if T is any tree on m vertices and if $m-1$ divides $n-1$, then $r(T, K_{1,n}) = m+n-1$;

(*c*)* if T is any tree on m vertices, then $r(T, K_n) = (m-1)(n-1)+1$.

(V. Chvátal)

7.3 TURÁN'S THEOREM

In this section, we shall prove a well-known theorem due to Turán (1941). It determines the maximum number of edges that a simple graph on ν vertices can have without containing a clique of size $m+1$. Turán's theorem has become the basis of a significant branch of graph theory known as *extremal graph theory* (see Erdös, 1967). We shall derive it from the following result of Erdös (1970).

Theorem 7.8 If a simple graph G contains no K_{m+1}, then G is degree-majorised by some complete m-partite graph H. Moreover, if G has the same degree sequence as H, then $G \cong H$.

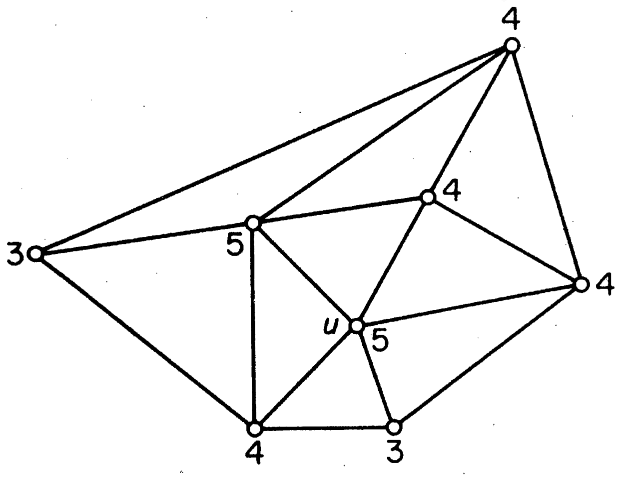
Proof By induction on m. The theorem is trivial for $m = 1$. Assume that it holds for all $m < n$, and let G be a simple graph which contains no K_{n+1}. Choose a vertex u of degree Δ in G, and set $G_1 = G[N(u)]$. Since G contains no K_{n+1}, G_1 contains no K_n and therefore, by the induction hypothesis, is degree-majorised by some complete $(n-1)$-partite graph H_1.

Next, set $V_1 = N(u)$ and $V_2 = V \backslash V_1$, and denote by G_2 the graph whose vertex set is V_2 and whose edge set is empty. Consider the join $G_1 \vee G_2$ of G_1 and G_2. Since

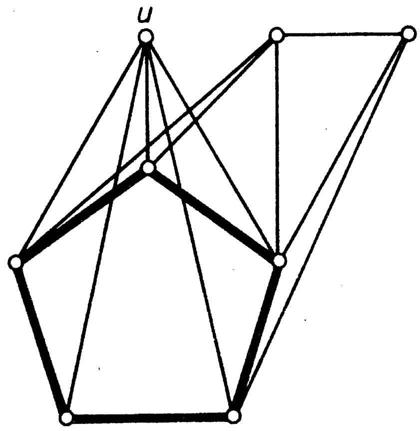
$$N_G(v) \subseteq N_{G_1 \vee G_2}(v) \quad \text{for} \quad v \in V_1 \tag{7.15}$$

and since each vertex of V_2 has degree Δ in $G_1 \vee G_2$, G is degree-majorised by $G_1 \vee G_2$. Therefore G is also degree-majorised by the complete n-partite graph $H = H_1 \vee G_2$. (See figure 7.3 for illustration.)

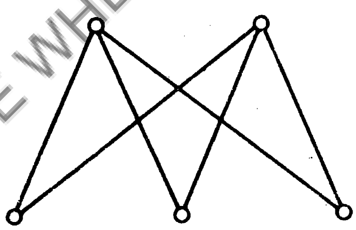
5689



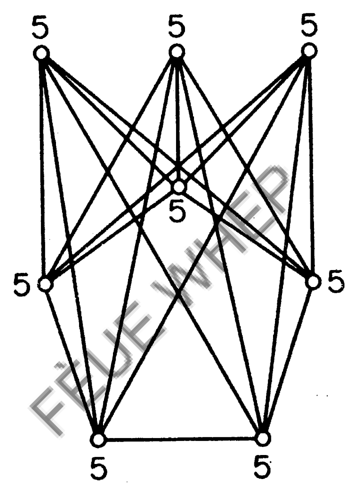
$G(3,3,4,4,4,4,5,5)$



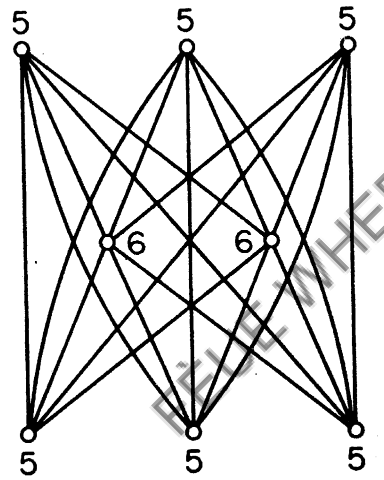
Another diagram of G with $G_1 = G[N(u)]$ indicated



H_1



$G_1 \vee G_2$ (5,5,5,5,5,5,5,5)



$H = H_1 \vee G_2$ (5,5,5,5,5,5,6,6)

Figure 7.3

Suppose, now, that G has the same degree sequence as H. Then G has the same degree sequence as $G_1 \vee G_2$ and hence equality must hold in (7.15). Thus, in G, every vertex of V_1 must be joined to every vertex of V_2. It follows that $G = G_1 \vee G_2$. Since $G = G_1 \vee G_2$ has the same degree sequence as $H = H_1 \vee G_2$, the graphs G_1 and H_1 must have the same degree sequence and therefore, by the induction hypothesis, be isomorphic. We conclude that $G \cong H$ □

It is interesting to note that the above theorem bears a striking similarity to theorem 4.6.

Let $T_{m,n}$ denote the complete m-partite graph on n vertices in which all parts are as equal in size as possible; the graph H of figure 7.3 is $T_{3,8}$.

Theorem 7.9 If G is simple and contains no K_{m+1}, then $\varepsilon(G) \leq \varepsilon(T_{m,\nu})$. Moreover, $\varepsilon(G) = \varepsilon(T_{m,\nu})$ only if $G \cong T_{m,\nu}$.

FEUE WHEP

5690

Proof Let G be a simple graph that contains no K_{m+1}. By theorem 7.8, G is degree-majorised by some complete m-partite graph H. It follows from theorem 1.1 that

$$\varepsilon(G) \le \varepsilon(H) \tag{7.16}$$

But (exercise 1.2.9)

$$\varepsilon(H) \le \varepsilon(T_{m,\nu}) \tag{7.17}$$

Therefore, from (7.16) and (7.17)

$$\varepsilon(G) \le \varepsilon(T_{m,\nu}) \tag{7.18}$$

proving the first assertion.

Suppose, now, that equality holds in (7.18). Then equality must hold in both (7.16) and (7.17). Since $\varepsilon(G) = \varepsilon(H)$ and G is degree-majorised by H, G must have the same degree sequence as H. Therefore, by theorem 7.8, $G \cong H$. Also, since $\varepsilon(H) = \varepsilon(T_{m,\nu})$, it follows (exercise 1.2.9) that $H \cong T_{m,\nu}$. We conclude that $G \cong T_{m,\nu}$ □

Exercises

7.3.1 In a group of nine people, one person knows two of the others, two people each know four others, four each know five others, and the remaining two each know six others. Show that there are three people who all know one another.

7.3.2 A certain bridge club has a special rule to the effect that four members may play together only if no two of them have previously partnered one another. At one meeting fourteen members, each of whom has previously partnered five others, turn up. Three games are played, and then proceedings come to a halt because of the club rule. Just as the members are preparing to leave, a new member, unknown to any of them, arrives. Show that at least one more game can now be played.

7.3.3 (*a*) Show that if G is simple and $\varepsilon > \nu^2/4$, then G contains a triangle.

(*b*) Find a simple graph G with $\varepsilon = [\nu^2/4]$ that contains no triangle.

(*c*)* Show that if G is simple and not bipartite with $\varepsilon > ((\nu-1)^2/4)+1$, then G contains a triangle.

(*d*) Find a simple non-bipartite graph G with $\varepsilon = [(\nu-1)^2/4]+1$ that contains no triangle. (P. Erdös)

7.3.4 (*a*)* Show that if G is simple and $\sum_{v\in V}\binom{d(v)}{2} > (m-1)\binom{\nu}{2}$, then G contains $K_{2,m}(m \ge 2)$.

(*b*) Deduce that if G is simple and $\varepsilon > \dfrac{(m-1)^{\frac{1}{2}}\nu^{\frac{3}{2}}}{2} + \dfrac{\nu}{4}$, then G contains $K_{2,m}(m \ge 2)$.

FEUE WHEP

5691

(*c*) Show that, given a set of n points in the plane, the number of pairs of points at distance exactly 1 is at most $n^{\frac{3}{2}}/\sqrt{2}+n/4$.

7.3.5 Show that if G is simple and $\varepsilon > \dfrac{(m-1)^{1/m}\nu^{2-1/m}}{2} + \dfrac{(m-1)\nu}{2}$ then G contains $K_{m,m}$.

APPLICATIONS

7.4 SCHUR'S THEOREM

Consider the partition $(\{1, 4, 10, 13\}, \{2, 3, 11, 12\}, \{5, 6, 7, 8, 9\})$ of the set of integers $\{1, 2, \ldots, 13\}$. We observe that in no subset of the partition are there integers x, y and z (not necessarily distinct) which satisfy the equation

$$x+y=z \tag{7.19}$$

Yet, no matter how we partition $\{1, 2, \ldots, 14\}$ into three subsets, there always exists a subset of the partition which contains a solution to (7.19). Schur (1916) proved that, in general, given any positive integer n, there exists an integer f_n such that, in any partition of $\{1, 2, \ldots, f_n\}$ into n subsets, there is a subset which contains a solution to (7.19). We shall show how Schur's theorem follows from the existence of the Ramsey numbers r_n (defined in exercise 7.2.3).

Theorem 7.10 Let $(S_1, S_2, \ldots, S_n)$ be any partition of the set of integers $\{1, 2, \ldots, r_n\}$. Then, for some i, S_i contains three integers x, y and z satisfying the equation $x+y=z$.

Proof Consider the complete graph whose vertex set is $\{1, 2, \ldots, r_n\}$. Colour the edges of this graph in colours $1, 2, \ldots, n$ by the rule that the edge uv is assigned colour j if and only if $|u-v| \in S_j$. By Ramsey's theorem (7.7) there exists a monochromatic triangle; that is, there are three vertices a, b and c such that ab, bc and ca have the same colour, say i. Assume, without loss of generality that $a > b > c$ and write $x=a-b$, $y=b-c$ and $z=a-c$. Then $x, y, z \in S_i$ and $x+y=z$ □

Let s_n denote the least integer such that, in any partition of $\{1, 2, \ldots, s_n\}$ into n subsets, there is a subset which contains a solution to (7.19). It can be easily seen that $s_1=2$, $s_2=5$ and $s_3=14$ (exercise 7.4.1). Also, from theorem 7.10 and exercise 7.2.3 we have the upper bound

$$s_n \le r_n \le [n!\,e]+1$$

Exercise 7.4.2*b* provides a lower bound for s_n.

FEUE WHEP

5692

Exercises

7.4.1 Show that $s_1 = 2$, $s_2 = 5$ and $s_3 = 14$.

7.4.2 (*a*) Show that $s_n \geq 3s_{n-1} - 1$.

(*b*) Using (*a*) and the fact that $s_3 = 14$, show that $s_n \geq \frac{1}{2}(27(3)^{n-3} + 1)$. (A better lower bound has been obtained by Abbott and Moser, 1966.)

7.5 A GEOMETRY PROBLEM

The *diameter* of a set S of points in the plane is the maximum distance between two points of S. It should be noted that this is a purely geometric notion and is quite unrelated to the graph-theoretic concepts of diameter and distance.

We shall discuss sets of diameter 1. A set of n points determines $\binom{n}{2}$ distances between pairs of these points. It is intuitively clear that if n is 'large', then some of these distances must be 'small'. Therefore, for any d between 0 and 1, we can ask how many pairs of points in a set $\{x_1, x_2, \ldots, x_n\}$ of diameter 1 can be at distance greater than d. Here, we shall present a solution to one special case of this problem, namely when $d = 1/\sqrt{2}$.

As an illustration, consider the case $n = 6$. We then have six points x_1, x_2, x_3, x_4, x_5 and x_6. If we place them at the vertices of a regular hexagon so that the pairs (x_1, x_4), (x_2, x_5) and (x_3, x_6) are at distance 1, as shown in figure 7.4*a*, these six points constitute a set of diameter 1.

It is easily calculated that the pairs (x_1, x_2), (x_2, x_3), (x_3, x_4), (x_4, x_5), (x_5, x_6) and (x_6, x_1) are at distance 1/2, and the pairs (x_1, x_3), (x_2, x_4), (x_3, x_5), (x_4, x_6), (x_5, x_1) and (x_6, x_2) are at distance $\sqrt{3}/2$. Since $\sqrt{3}/2 > \sqrt{2}/2 = 1/\sqrt{2}$, there are nine pairs of points at distance greater than $1/\sqrt{2}$ in this set of diameter 1.

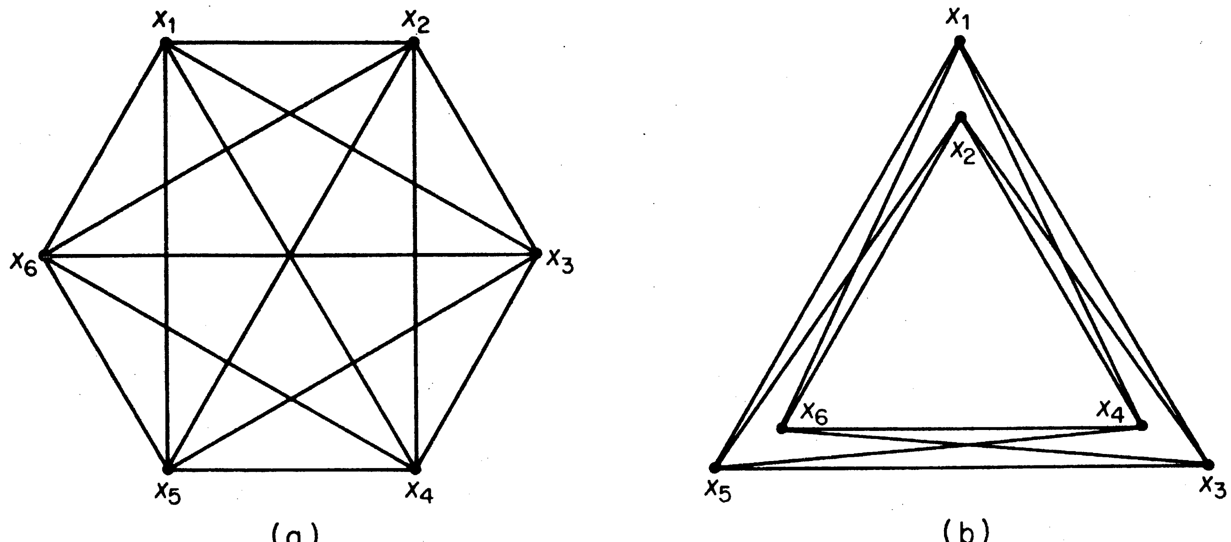


Figure 7.4

5693

However, nine is not the best that we can do with six points. By placing the points in the configuration shown in figure 7.4*b*, all pairs of points except (x_1, x_2), (x_3, x_4) and (x_5, x_6) are at distance greater than $1/\sqrt{2}$. Thus we have twelve pairs at distance greater than $1/\sqrt{2}$; this is, in fact, the best we can do. The solution to the problem in general is given by the following theorem.

Theorem 7.11 If $\{x_1, x_2, \ldots, x_n\}$ is a set of diameter 1 in the plane, the maximum possible number of pairs of points at distance greater than $1/\sqrt{2}$ is $[n^2/3]$. Moreover, for each n, there is a set $\{x_1, x_2, \ldots, x_n\}$ of diameter 1 with exactly $[n^2/3]$ pairs of points at distance greater than $1/\sqrt{2}$.

Proof Let G be the graph defined by

$$V(G) = \{x_1, x_2, \ldots, x_n\}$$

and

$$E(G) = \{x_i x_j \mid d(x_i, x_j) > 1/\sqrt{2}\}$$

where $d(x_i, x_j)$ here denotes the *euclidean* distance between x_i and x_j. We shall show that G cannot contain a K_4.

First, note that any four points in the plane must determine an angle of at least 90°. For the convex hull of the points is either (a) a line, (b) a triangle, or (c) a quadrilateral (see figure 7.5). Clearly, in each case there is an angle $x_i x_j x_k$ of at least 90°.

Now look at the three points x_i, x_j, x_k which determine this angle. Not all the distances $d(x_i, x_j)$, $d(x_i, x_k)$ and $d(x_j, x_k)$ can be greater than $1/\sqrt{2}$ and less than or equal to 1. For, if $d(x_i, x_j) > 1/\sqrt{2}$ and $d(x_j, x_k) > 1/\sqrt{2}$, then $d(x_i, x_k) > 1$. Since the set $\{x_1, x_2, \ldots, x_n\}$ is assumed to have diameter 1, it follows that, of any four points in G, at least one pair cannot be joined by an edge, and hence that G cannot contain a K_4. By Turán's theorem (7.9)

$$\varepsilon(G) \leq \varepsilon(T_{3,n}) = [n^2/3]$$

One can construct a set $\{x_1, x_2, \ldots, x_n\}$ of diameter 1 in which exactly

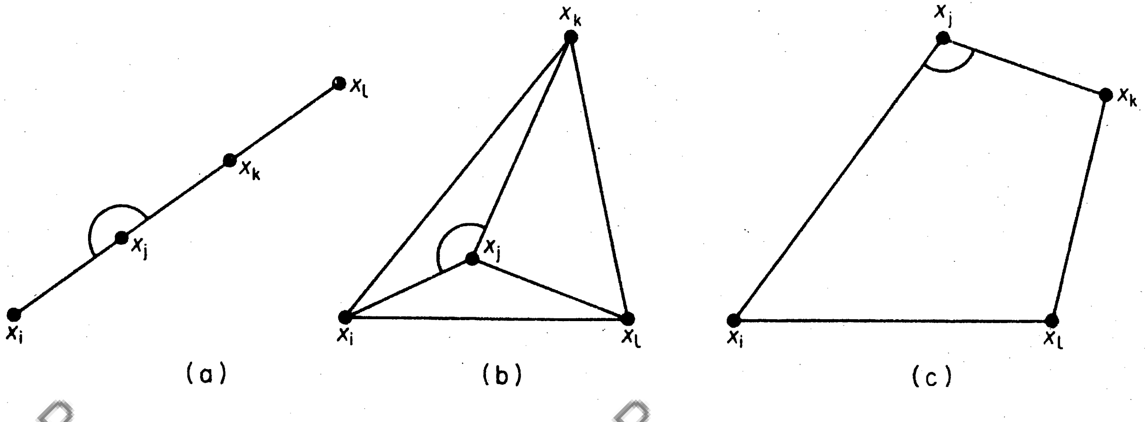


Figure 7.5

FEUE WHEP

5694

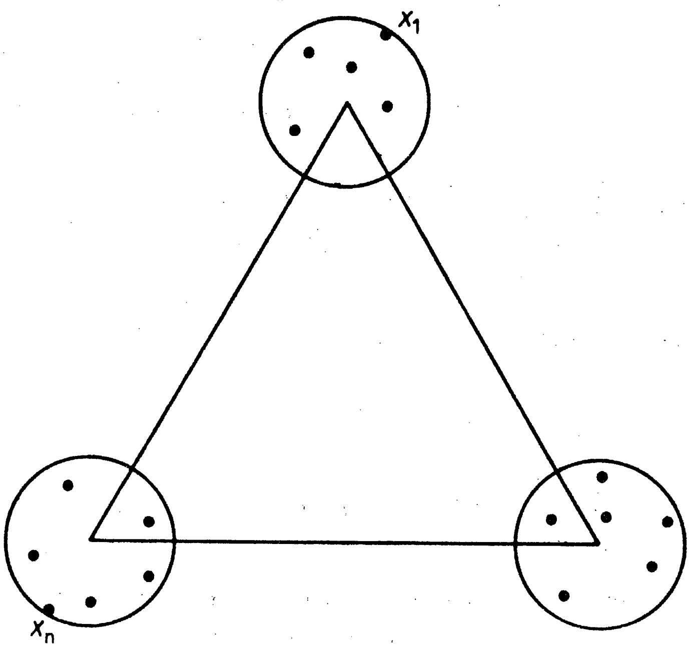


Figure 7.6

$[n^2/3]$ pairs of points are at distance greater than $1/\sqrt{2}$ as follows. Choose r such that $0<r<(1-1/\sqrt{2})/4$, and draw three circles of radius r whose centres are at a distance of $1-2r$ from one another (figure 7.6). Place $x_1, \ldots, x_{[n/3]}$ in one circle, $x_{[n/3]+1}, \ldots, x_{[2n/3]}$ in another, and $x_{[2n/3]+1}, \ldots, x_n$ in the third, in such a way that $d(x_1, x_n)=1$. This set clearly has diameter 1. Also, $d(x_i, x_j)>1/\sqrt{2}$ if and only if x_i and x_j are in different circles, and so there are exactly $[n^2/3]$ pairs (x_i, x_j) for which $d(x_i, x_j)>1/\sqrt{2}$ □

Exercises

7.5.1* Let $\{x_1, x_2, \ldots, x_n\}$ be a set of diameter 1 in the plane.

(*a*) Show that the maximum possible number of pairs of points at distance 1 is n.

(*b*) Construct a set $\{x_1, x_2, \ldots, x_n\}$ of diameter 1 in the plane in which exactly n pairs of points are at distance 1. (E. Pannwitz)

7.5.2 A flat circular city of radius six miles is patrolled by eighteen police cars, which communicate with one another by radio. If the range of a radio is nine miles, show that, at any time, there are always at least two cars each of which can communicate with at least five other cars.

REFERENCES

Abbott, H. L. (1972). Lower bounds for some Ramsey numbers. *Discrete Math.*, **2,** 289–93

Abbott, H. L. and Moser, L. (1966). Sum-free sets of integers. *Acta Arith.*, **11,** 392–96

5695

Erdös, P. (1947). Some remarks on the theory of graphs. *Bull. Amer. Math. Soc.*, **53,** 292–94

Erdös, P. (1967). Extremal problems in graph theory, in *A Seminar on Graph Theory* (ed. F. Harary), Holt, Rinehart and Winston, New York, pp. 54–59

Erdös, P. (1970). On the graph-theorem of Turán (Hungarian). *Mat. Lapok*, **21,** 249–51

Erdös, P. and Spencer, J. (1974). *Probabilistic Methods in Combinatorics*, Akadémiai Kiadó, Budapest

Erdös, P. and Szekeres, G. (1935). A combinatorial problem in geometry. *Compositio Math.*, **2,** 463–70

Folkman, J. (1970). Graphs with monochromatic complete subgraphs in every edge coloring. *SIAM J. Appl. Math.*, **18,** 19–24

Gallai, T. (1959). Über extreme Punkt- und Kantenmengen. *Ann. Univ. Sci. Budapest, Eötvös Sect. Math.*, **2,** 133–38

Greenwood, R. E. and Gleason, A. M. (1955). Combinatorial relations and chromatic graphs. *Canad. J. Math.*, **7,** 1–7

Ramsey, F. P. (1930). On a problem of formal logic. *Proc. London Math. Soc.*, **30,** 264–86

Schur, I. (1916). Über die Kongruenz $x^m + y^m = z^m$ (mod p). *Jber. Deutsch. Math.-Verein.*, **25,** 114–17

Turán, P. (1941). An extremal problem in graph theory (Hungarian). *Mat. Fiz. Lapok*, **48,** 436–52

5696

8 Vertex Colourings

8.1 CHROMATIC NUMBER

In chapter 6 we studied edge colourings of graphs. We now turn our attention to the analogous concept of vertex colouring.

A *k-vertex colouring* of G is an assignment of k colours, $1, 2, \ldots, k$, to the vertices of G; the colouring is *proper* if no two distinct adjacent vertices have the same colour. Thus a proper k-vertex colouring of a loopless graph G is a partition $(V_1, V_2, \ldots, V_k)$ of V into k (possibly empty) independent sets. G is *k-vertex-colourable* if G has a proper k-vertex colouring. It will be convenient to refer to a 'proper vertex colouring' as, simply, a *colouring* and to a 'proper k-vertex colouring' as a *k-colouring;* we shall similarly abbreviate 'k-vertex-colourable' to *k-colourable.* Clearly, a graph is k-colourable if and only if its underlying simple graph is k-colourable. Therefore, in discussing colourings, we shall restrict ourselves to simple graphs; a simple graph is 1-colourable if and only if it is empty, and 2-colourable if and only if it is bipartite. The *chromatic number*, $\chi(G)$, of G is the minimum k for which G is k-colourable; if $\chi(G) = k$, G is said to be *k-chromatic.* A 3-chromatic graph is shown in figure 8.1. It has the indicated 3-colouring, and is not 2-colourable since it is not bipartite.

It is helpful, when dealing with colourings, to study the properties of a special class of graphs called critical graphs. We say that a graph G is *critical* if $\chi(H) < \chi(G)$ for every proper subgraph H of G. Such graphs were first investigated by Dirac (1952). A *k-critical* graph is one that is k-chromatic and critical; every k-chromatic graph has a k-critical subgraph. A 4-critical graph, due to Grötzsch (1958), is shown in figure 8.2.

An easy consequence of the definition is that every critical graph is connected. The following theorems establish some of the basic properties of critical graphs.

Theorem 8.1 If G is k-critical, then $\delta \geq k-1$.

Proof By contradiction. If possible, let G be a k-critical graph with $\delta < k-1$, and let v be a vertex of degree δ in G. Since G is k-critical, $G - v$ is $(k-1)$-colourable. Let $(V_1, V_2, \ldots, V_{k-1})$ be a $(k-1)$-colouring of $G - v$. By definition, v is adjacent in G to $\delta < k-1$ vertices, and therefore v must be nonadjacent in G to every vertex of some V_j. But then $(V_1, V_2, \ldots, V_j \cup \{v\}, \ldots, V_{k-1})$ is a $(k-1)$-colouring of G, a contradiction. Thus $\delta \geq k-1$ □

5697

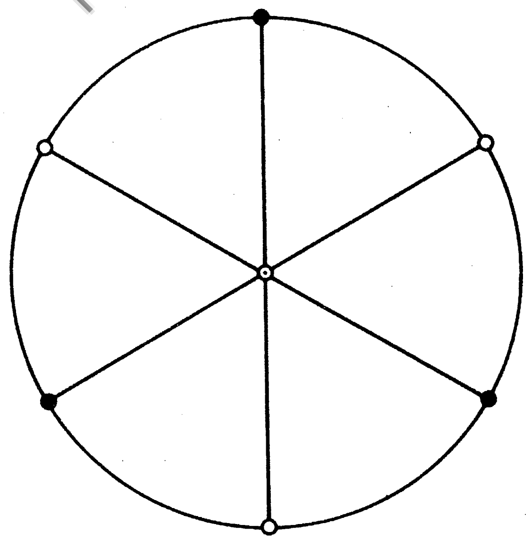


Figure 8.1. A 3-chromatic graph

Corollary 8.1.1 Every k-chromatic graph has at least k vertices of degree at least $k-1$.

Proof Let G be a k-chromatic graph, and let H be a k-critical subgraph of G. By theorem 8.1, each vertex of H has degree at least $k-1$ in H, and hence also in G. The corollary now follows since H, being k-chromatic, clearly has at least k vertices □

Corollary 8.1.2 For any graph G,

$$\chi \leq \Delta + 1$$

Proof This is an immediate consequence of corollary 8.1.1 □

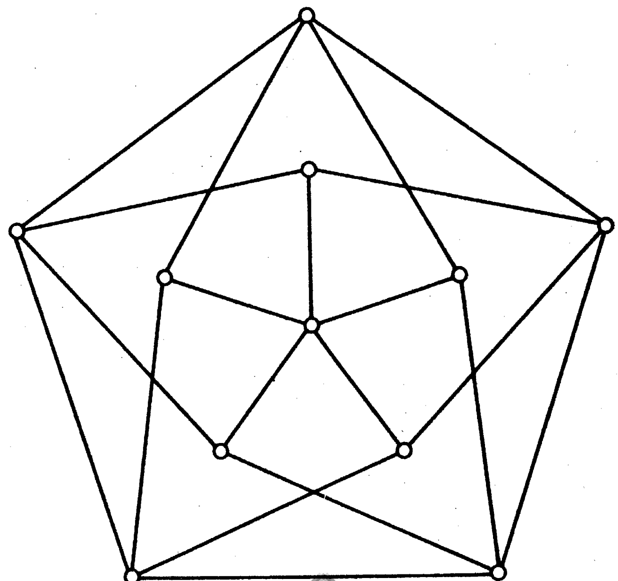


Figure 8.2. The Grötzsch graph—a 4-critical graph

FEUE WHEP

Let S be a vertex cut of a connected graph G, and let the components of $G-S$ have vertex sets $V_1, V_2, \ldots, V_n$. Then the subgraphs $G_i = G[V \cup S]$ are called the *S-components* of G (see figure 8.3). We say that colourings of $G_1, G_2, \ldots, G_n$ *agree* on S if, for every $v \in S$, vertex v is assigned the same colour in each of the colourings.

Theorem 8.2 In a critical graph, no vertex cut is a clique.

Proof By contradiction. Let G be a k-critical graph, and suppose that G has a vertex cut S that is a clique. Denote the S-components of G by $G_1, G_2, \ldots, G_n$. Since G is k-critical, each G_i is $(k-1)$-colourable. Furthermore, because S is a clique, the vertices in S must receive distinct colours in any $(k-1)$-colouring of G_i. It follows that there are $(k-1)$-colourings of $G_1, G_2, \ldots, G_n$ which agree on S. But these colourings together yield a $(k-1)$-colouring of G, a contradiction □

Corollary 8.2 Every critical graph is a block.

Proof If v is a cut vertex, then $\{v\}$ is a vertex cut which is also, trivially, a clique. It follows from theorem 8.2 that no critical graph has a cut vertex; equivalently, every critical graph is a block □

Another consequence of theorem 8.2 is that if a k-critical graph G has a 2-vertex cut $\{u, v\}$, then u and v cannot be adjacent. We shall say that a $\{u, v\}$-component G_i of G is of *type 1* if every $(k-1)$-colouring of G_i assigns the same colour to u and v, and of *type 2* if every $(k-1)$-colouring of G_i assigns different colours to u and v (see figure 8.4).

Theorem 8.3 (Dirac, 1953) Let G be a k-critical graph with a 2-vertex cut $\{u, v\}$. Then

(i) $G = G_1 \cup G_2$, where G_i is a $\{u, v\}$-component of type i ($i = 1, 2$), and

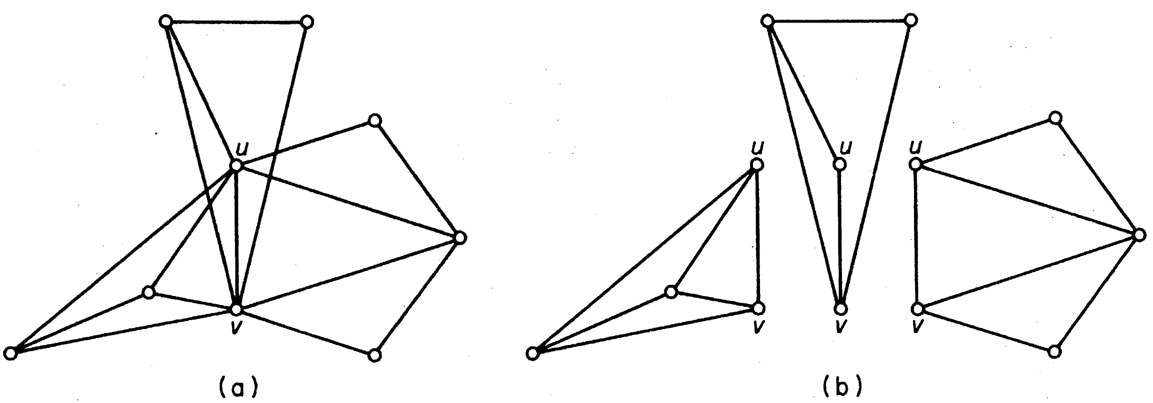


Figure 8.3. (*a*) G; (*b*) the $\{u, v\}$-components of G

FEUE WHEP

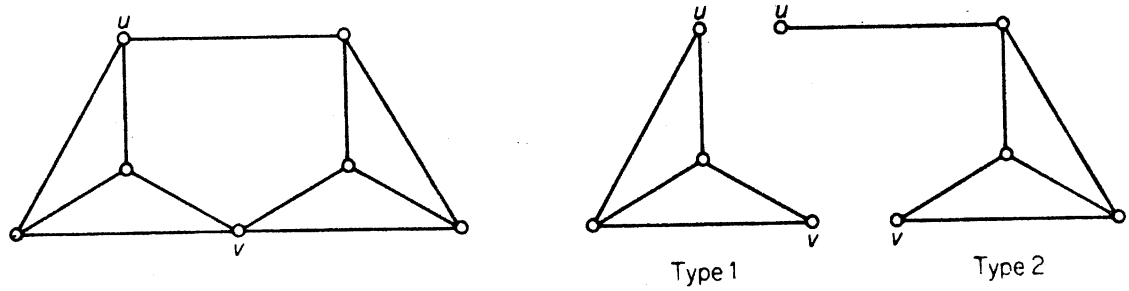


Figure 8.4

(ii) both $G_1 + uv$ and $G_2 \cdot uv$ are k-critical (where $G_2 \cdot uv$ denotes the graph obtained from G_2 by identifying u and v).,

Proof (i) Since G is critical, each $\{u, v\}$-component of G is $(k-1)$-colourable. Now there cannot exist $(k-1)$-colourings of these $\{u, v\}$-components all of which agree on $\{u, v\}$, since such colourings would together yield a $(k-1)$-colouring of G. Therefore there are two $\{u, v\}$-components G_1 and G_2 such that no $(k-1)$-colouring of G_1 agrees with any $(k-1)$-colouring of G_2. Clearly one, say G_1, must be of type 1 and the other, G_2, of type 2. Since G_1 and G_2 are of different types, the subgraph $G_1 \cup G_2$ of G is not $(k-1)$-colourable. Therefore, because G is critical, we must have $G = G_1 \cup G_2$.

(ii) Set $H_1 = G_1 + uv$. Since G_1 is of type 1, H_1 is k-chromatic. We shall prove that H_1 is critical by showing that, for every edge e of H_1, $H_1 - e$ is $(k-1)$-colourable. This is clearly so if $e = uv$, since then $H_1 - e = G_1$. Let e be some other edge of H_1. In any $(k-1)$-colouring of $G - e$, the vertices u and v must receive different colours, since G_2 is a subgraph of $G - e$. The restriction of such a colouring to the vertices of G_1 is a $(k-1)$-colouring of $H_1 - e$. Thus $G_1 + uv$ is k-critical. An analogous argument shows that $G_2 \cdot uv$ is k-critical □

Corollary 8.3 Let G be a k-critical graph with a 2-vertex cut $\{u, v\}$. Then

$$d(u) + d(v) \geq 3k - 5 \tag{8.1}$$

Proof Let G_1 be the $\{u, v\}$-component of type 1 and G_2 the $\{u, v\}$-component of type 2. Set $H_1 = G_1 + uv$ and $H_2 = G_2 \cdot uv$. By theorems 8.3 and 8.1

$$d_{H_1}(u) + d_{H_1}(v) \geq 2k - 2$$

and

$$d_{H_2}(w) \geq k - 1$$

where w is the new vertex obtained by identifying u and v.

It follows that

$$d_{G_1}(u) + d_{G_1}(v) \geq 2k - 4$$

5700

and

$$d_{G_2}(u) + d_{G_2}(v) \geq k - 1$$

These two inequalities yield (8.1) □

Exercises

8.1.1 Show that if G is simple, then $\chi \geq \nu^2/(\nu^2 - 2\varepsilon)$.

8.1.2 Show that if any two odd cycles of G have a vertex in common, then $\chi \leq 5$.

8.1.3 Show that if G has degree sequence $(d_1, d_2, \ldots, d_\nu)$ with $d_1 \geq d_2 \geq \ldots \geq d_\nu$, then $\chi \leq \max\limits_i \min\{d_i + 1, i\}$.

(D. J. A. Welsh and M. B. Powell)

8.1.4 Using exercise 8.1.3, show that

(*a*) $\chi \leq \{(2\varepsilon)^{\frac{1}{2}}\}$;

(*b*) $\chi(G) + \chi(G^c) \leq \nu + 1$. (E. A. Nordhaus and J. W. Gaddum)

8.1.5 Show that $\chi(G) \leq 1 + \max \delta(H)$, where the maximum is taken over all induced subgraphs H of G. (G. Szekeres and H. S. Wilf)

8.1.6* If a k-chromatic graph G has a colouring in which each colour is assigned to at least two vertices, show that G has a k-colouring of this type. (T. Gallai)

8.1.7 Show that the only 1-critical graph is K_1, the only 2-critical graph is K_2, and the only 3-critical graphs are the odd k-cycles with $k \geq 3$.

8.1.8 A graph G is *uniquely k-colourable* if any two k-colourings of G induce the same partition of V. Show that no vertex cut of a k-critical graph induces a uniquely $(k-1)$-colourable subgraph.

8.1.9 (*a*) Show that if u and v are two vertices of a critical graph G, then $N(u) \not\subseteq N(v)$.

(*b*) Deduce that no k-critical graph has exactly $k+1$ vertices.

8.1.10 Show that

(*a*) $\chi(G_1 \vee G_2) = \chi(G_1) + \chi(G_2)$;

(*b*) $G_1 \vee G_2$ is critical if and only if both G_1 and G_2 are critical.

8.1.11 Let G_1 and G_2 be two k-critical graphs with exactly one vertex v in common, and let vv_1 and vv_2 be edges of G_1 and G_2. Show that the graph $(G_1 - vv_1) \cup (G_2 - vv_2) + v_1v_2$ is k-critical. (G. Hajós)

8.1.12 For $n = 4$ and all $n \geq 6$, construct a 4-critical graph on n vertices.

8.1.13 (*a*)* Let (X, Y) be a partition of V such that $G[X]$ and $G[Y]$ are both n-colourable. Show that, if the edge cut $[X, Y]$ has at most $n-1$ edges, then G is also n-colourable.

(P. C. Kainen)

(*b*) Deduce that every k-critical graph is $(k-1)$-edge-connected.

(G. A. Dirac)

FEUE WHEP

5701

8.2 BROOKS' THEOREM

The upper bound on chromatic number given in corollary 8.1.2 is sometimes very much greater than the actual value. For example, bipartite graphs are 2-chromatic, but can have arbitrarily large maximum degree. In this sense corollary 8.1.2 is a considerably weaker result than Vizing's theorem (6.2). There is another sense in which Vizing's result is stronger. Many graphs G satisfy $\chi' = \Delta + 1$ (see exercises 6.2.2 and 6.2.3). However, as is shown in the following theorem due to Brooks (1941), there are only two types of graph G for which $\chi = \Delta + 1$. The proof of Brooks' theorem given here is by Lovász (1973).

Theorem 8.4 If G is a connected simple graph and is neither an odd cycle nor a complete graph, then $\chi \leq \Delta$.

Proof Let G be a k-chromatic graph which satisfies the hypothesis of the theorem. Without loss of generality, we may assume that G is k-critical. By corollary 8.2, G is a block. Also, since 1-critical and 2-critical graphs are complete and 3-critical graphs are odd cycles (exercise 8.1.7), we have $k \geq 4$.

If G has a 2-vertex cut $\{u, v\}$, corollary 8.3 gives

$$2\Delta \geq d(u) + d(v) \geq 3k - 5 \geq 2k - 1$$

This implies that $\chi = k \leq \Delta$, since 2Δ is even.

Assume, then, that G is 3-connected. Since G is not complete, there are three vertices u, v and w in G such that $uv, vw \in E$ and $uw \notin E$ (exercise 1.6.14). Set $u = v_1$ and $w = v_2$ and let $v_3, v_4, \ldots, v_\nu = v$ be any ordering of the vertices of $G - \{u, w\}$ such that each v_i is adjacent to some v_j with $j > i$. (This can be achieved by arranging the vertices of $G - \{u, w\}$ in nonincreasing order of their distance from v.) We can now describe a Δ-colouring of G: assign colour 1 to $v_1 = u$ and $v_2 = w$; then successively colour $v_3, v_4, \ldots, v_\nu$, each with the first available colour in the list $1, 2, \ldots, \Delta$. By the construction of the sequence $v_1, v_2, \ldots, v_\nu$, each vertex v_i, $1 \leq i \leq \nu - 1$, is adjacent to some vertex v_j with $j > i$, and therefore to at most $\Delta - 1$ vertices v_j with $j < i$. It follows that, when its turn comes to be coloured, v_i is adjacent to at most $\Delta - 1$ colours, and thus that one of the colours $1, 2, \ldots, \Delta$ will be available. Finally, since v_ν is adjacent to two vertices of colour 1 (namely v_1 and v_2), it is adjacent to at most $\Delta - 2$ other colours and can be assigned one of the colours $2, 3, \ldots, \Delta$ □

Exercises

8.2.1 Show that Brooks' theorem is equivalent to the following statement: if G is k-critical ($k \geq 4$) and not complete, then $2\varepsilon \geq \nu(k-1)+1$.

FÊUE WHEP

5702

8.2.2 Use Brooks' theorem to show that if G is loopless with $\Delta=3$, then $\chi' \leq 4$.

8.3 HAJÓS' CONJECTURE

A *subdivision* of a graph G is a graph that can be obtained from G by a sequence of edge subdivisions. A subdivision of K_4 is shown in figure 8.5. Although no necessary and sufficient condition for a graph to be k-chromatic is known when $k \geq 3$, a plausible necessary condition has been proposed by Hajós (1961): if G is k-chromatic, then G contains a subdivision of K_k. This is known as *Hajós' conjecture.* It should be noted that the condition is not sufficient; for example, a 4-cycle is a subdivision of K_3, but is not 3-chromatic.

For $k=1$ and $k=2$, the validity of Hajós' conjecture is obvious. It is also easily verified for $k=3$, because a 3-chromatic graph necessarily contains an odd cycle, and every odd cycle is a subdivision of K_3. Dirac (1952) settled the case $k=4$.

Theorem 8.5 If G is 4-chromatic, then G contains a subdivision of K_4.

Proof Let G be a 4-chromatic graph. Note that if some subgraph of G contains a subdivision of K_4, then so, too, does G. Without loss of generality, therefore, we may assume that G is critical, and hence that G is a block with $\delta \geq 3$. If $\nu=4$, then G is K_4 and the theorem holds trivially. We proceed by induction on ν. Assume the theorem true for all 4-chromatic graphs with fewer than n vertices, and let $\nu(G)=n>4$.

Suppose, first, that G has a 2-vertex cut $\{u, v\}$. By theorem 8.3, G has two $\{u, v\}$-components G_1 and G_2, where G_1+uv is 4-critical. Since $\nu(G_1+uv) < \nu(G)$, we can apply the induction hypothesis and deduce that G_1+uv

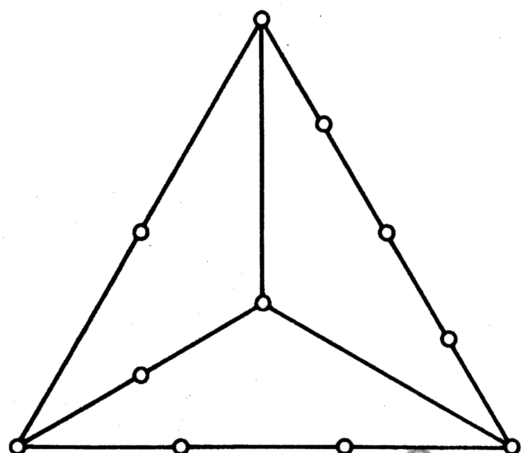


Figure 8.5. A subdivision of K_4

5703

contains a subdivision of K_4. It follows that, if P is a (u, v)-path in G_2, then $G_1 \cup P$ contains a subdivision of K_4. Hence so, too, does G, since $G_1 \cup P \subseteq G$.

Now suppose that G is 3-connected. Since $\delta \geq 3$, G has a cycle C of length at least four. Let u and v be nonconsecutive vertices on C. Since $G-\{u, v\}$ is connected, there is a path P in $G-\{u, v\}$ connecting the two components of $C-\{u, v\}$; we may assume that the origin x and the terminus y are the only vertices of P on C. Similarly, there is a path Q in $G-\{x, y\}$ (see figure 8.6).

If P and Q have no vertex in common, then $C \cup P \cup Q$ is a subdivision of K_4 (figure 8.6*a*). Otherwise, let w be the first vertex of P on Q, and let P' denote the (x, w)-section of P. Then $C \cup P' \cup Q$ is a subdivision of K_4 (figure 8.6*b*). Hence, in both cases, G contains a subdivision of K_4 □

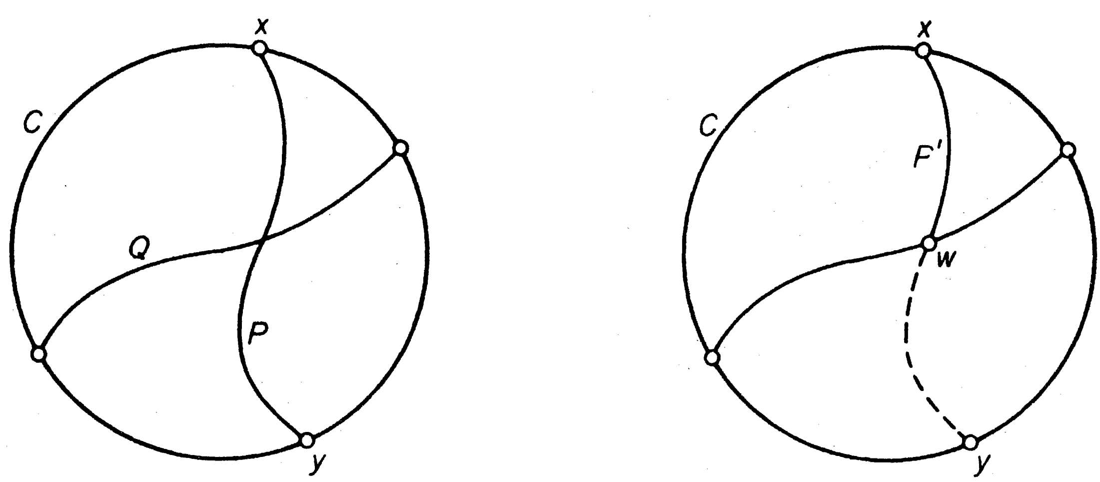
Hajós' conjecture has not yet been settled in general, and its resolution is known to be a very difficult problem. There is a related conjecture due to Hadwiger (1943): if G is k-chromatic, then G is 'contractible' to a graph which contains K_k. Wagner (1964) has shown that the case $k = 5$ of Hadwiger's conjecture is equivalent to the famous four-colour conjecture, to be discussed in chapter 9.

Exercises

8.3.1* Show that if G is simple and has at most one vertex of degree less than three, then G contains a subdivision of K_4.

8.3.2 (*a*)* Show that if G is simple with $\nu \geq 4$ and $\varepsilon \geq 2\nu - 2$, then G contains a subdivision of K_4.

(*b*) For $\nu \geq 4$, find a simple graph G with $\varepsilon = 2\nu - 3$ that contains no subdivision of K_4.



(a) (b)

Figure 8.6

5704

8.4 CHROMATIC POLYNOMIALS

In the study of colourings, some insight can be gained by considering not only the existence of colourings but the number of such colourings; this approach was developed by Birkhoff (1912) as a possible means of attacking the four-colour conjecture.

We shall denote the number of distinct k-colourings of G by $\pi_k(G)$; thus $\pi_k(G) > 0$ if and only if G is k-colourable. Two colourings are to be regarded as distinct if some vertex is assigned different colours in the two colourings; in other words, if $(V_1, V_2, \ldots, V_k)$ and $(V_1', V_2', \ldots, V_k')$ are two colourings, then $(V_1, V_2, \ldots, V_k) = (V_1', V_2', \ldots, V_k')$ if and only if $V_i = V_i'$ for $1 \le i \le k$. For example, a triangle has the six distinct 3-colourings shown in figure 8.7. Note that even though there is exactly one vertex of each colour in each colouring, we still regard these six colourings as distinct.

If G is empty, then each vertex can be independently assigned any one of the k available colours. Therefore $\pi_k(G) = k^\nu$. On the other hand, if G is complete, then there are k choices of colour for the first vertex, $k-1$ choices for the second, $k-2$ for the third, and so on. Thus, in this case, $\pi_k(G) = k(k-1)\ldots(k-\nu+1)$. In general, there is a simple recursion formula for $\pi_k(G)$. It bears a close resemblance to the recursion formula for $\tau(G)$ (the number of spanning trees of G), given in theorem 2.8.

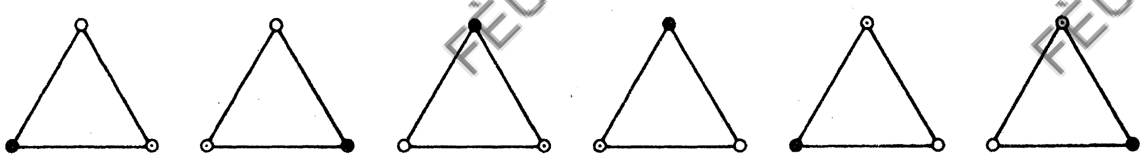


Figure 8.7

Theorem 8.6 If G is simple, then $\pi_k(G) = \pi_k(G-e) - \pi_k(G \cdot e)$ for any edge e of G.

Proof Let u and v be the ends of e. To each k-colouring of $G-e$ that assigns the same colour to u and v, there corresponds a k-colouring of $G \cdot e$ in which the vertex of $G \cdot e$ formed by identifying u and v is assigned the common colour of u and v. This correspondence is clearly a bijection (see figure 8.8). Therefore $\pi_k(G \cdot e)$ is precisely the number of k-colourings of $G-e$ in which u and v are assigned the same colour.

Also, since each k-colouring of $G-e$ that assigns different colours to u and v is a k-colouring of G, and conversely, $\pi_k(G)$ is the number of k-colourings of $G-e$ in which u and v are assigned different colours. It follows that $\pi_k(G-e) = \pi_k(G) + \pi_k(G \cdot e)$. □

FEUE WHEP

5705

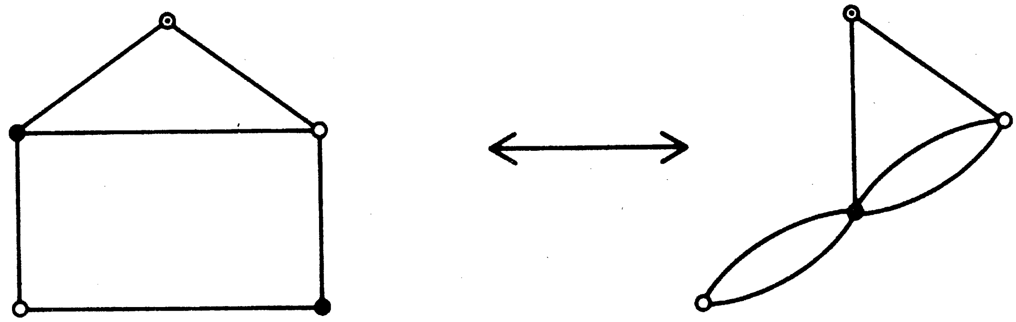


Figure 8.8

Corollary 8.6 For any graph G, $\pi_k(G)$ is a polynomial in k of degree ν, with integer coefficients, leading term k^ν and constant term zero. Furthermore, the coefficients of $\pi_k(G)$ alternate in sign.

Proof By induction on ε. We may assume, without loss of generality, that G is simple. If $\varepsilon = 0$ then, as has already been noted, $\pi_k(G) = k^\nu$, which trivially satisfies the conditions of the corollary. Suppose, now, that the corollary holds for all graphs with fewer than m edges, and let G be a graph with m edges, where $m \geq 1$. Let e be any edge of G. Then both $G - e$ and $G \cdot e$ have $m - 1$ edges, and it follows from the induction hypothesis that there are non-negative integers $a_1, a_2, \ldots, a_{\nu-1}$ and $b_1, b_2, \ldots, b_{\nu-2}$ such that

$$\pi_k(G-e) = \sum_{i=1}^{\nu-1} (-1)^{\nu-i} a_i k^i + k^\nu$$

and

$$\pi_k(G \cdot e) = \sum_{i=1}^{\nu-2} (-1)^{\nu-i-1} b_i k^i + k^{\nu-1}$$

By theorem 8.6

$$\begin{aligned}\pi_k(G) &= \pi_k(G-e) - \pi_k(G \cdot e)\\ &= \sum_{i=1}^{\nu-2} (-1)^{\nu-i}(a_i + b_i)k^i - (a_{\nu-1}+1)k^{\nu-1} + k^\nu\end{aligned}$$

Thus G, too, satisfies the conditions of the corollary. The result follows by the principle of induction □

By virtue of corollary 8.6, we can now refer to the function $\pi_k(G)$ as the *chromatic polynomial* of G. Theorem 8.6 provides a means of calculating the chromatic polynomial of a graph recursively. It can be used in either of two ways:

(i) by repeatedly applying the recursion $\pi_k(G) = \pi_k(G-e) - \pi_k(G \cdot e)$, and thereby expressing $\pi_k(G)$ as a linear combination of chromatic polynomials of empty graphs, or

(ii) by repeatedly applying the recursion $\pi_k(G-e) = \pi_k(G) + \pi_k(G \cdot e)$, and

FEUE WHEP

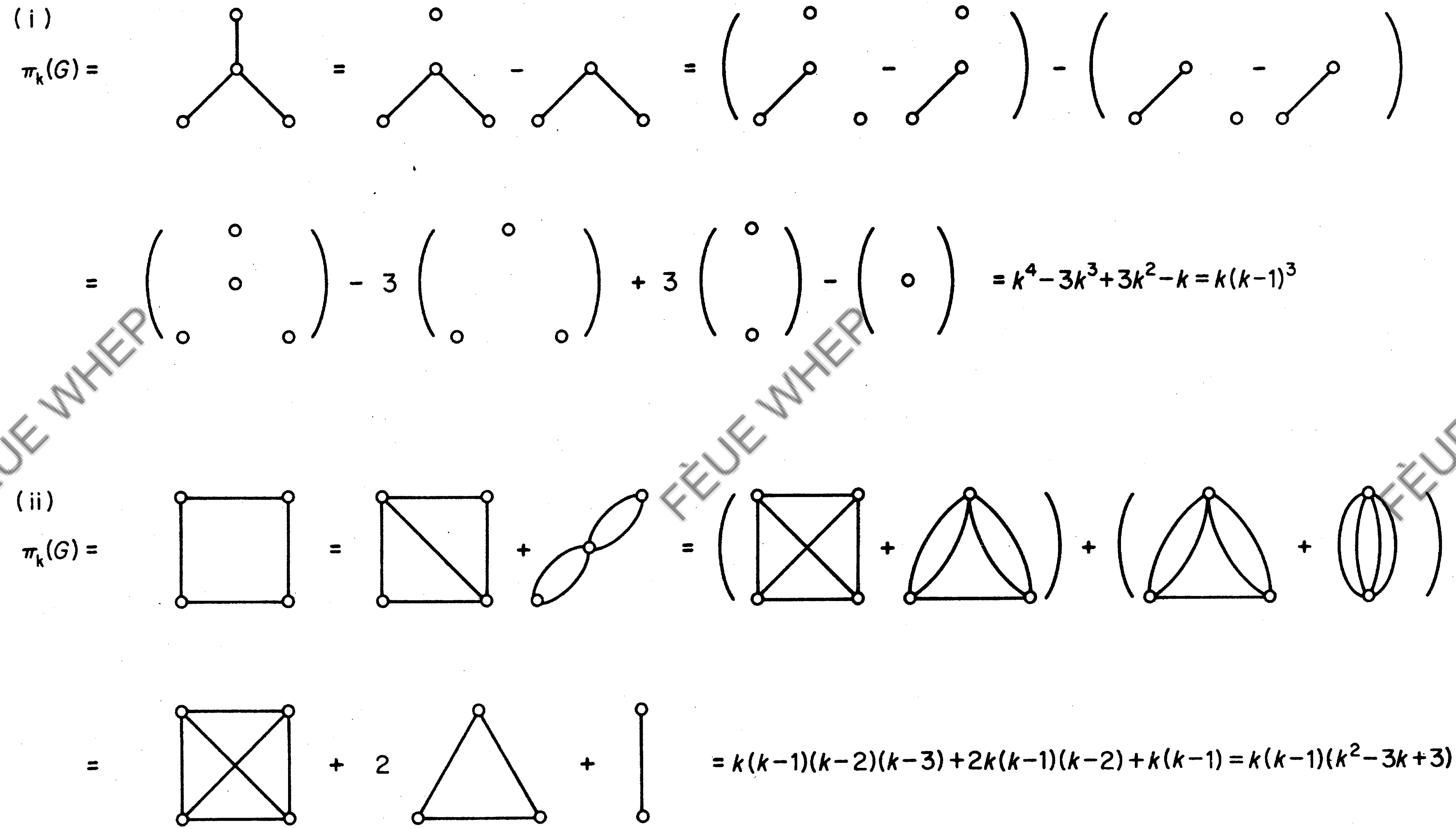


Figure 8.9. Recursive calculation of $\pi_k(G)$

FĚUE WHEP

5707

thereby expressing $\pi_k(G)$ as a linear combination of chromatic polynomials of complete graphs.

Method (i) is more suited to graphs with few edges, whereas (ii) can be applied more efficiently to graphs with many edges. These two methods are illustrated in figure 8.9 (where the chromatic polynomial of a graph is represented symbolically by the graph itself).

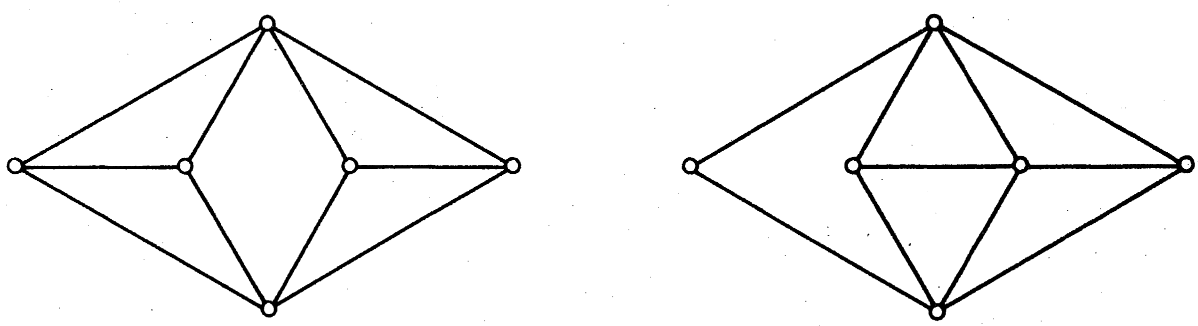
The calculation of chromatic polynomials can sometimes be facilitated by the use of a number of formulae relating the chromatic polynomial of G to the chromatic polynomials of various subgraphs of G (see exercises 8.4.5*a*, 8.4.6 and 8.4.7). However, no good algorithm is known for finding the chromatic polynomial of a graph. (Such an algorithm would clearly provide an efficient way to determine the chromatic number.)

Although many properties of chromatic polynomials are known, no one has yet discovered which polynomials are chromatic. It has been conjectured by Read (1968) that the sequence of coefficients of any chromatic polynomial must first rise in absolute value and then fall—in other words, that no coefficient may be flanked by two coefficients having greater absolute value. However, even if true, this condition, together with the conditions of corollary 8.6, would not be enough. The polynomial $k^4-3k^3+3k^2$, for example, satisfies all these conditions, but still is not the chromatic polynomial of any graph (exercise 8.4.2*b*).

Chromatic polynomials have been used with some success in the study of planar graphs, where their roots exhibit an unexpected regularity (see Tutte, 1970). Further results on chromatic polynomials can be found in the lucid survey article by Read (1968).

Exercises

8.4.1 Calculate the chromatic polynomials of the following two graphs:



8.4.2 (*a*) Show, by means of theorem 8.6, that if G is simple, then the coefficient of $k^{\nu-1}$ in $\pi_k(G)$ is $-\varepsilon$.

(*b*) Deduce that no graph has chromatic polynomial $k^4-3k^3+3k^2$.

8.4.3 (*a*) Show that if G is a tree, then $\pi_k(G)=k(k-1)^{\nu-1}$.

(*b*) Deduce that if G is connected, then $\pi_k(G)\le k(k-1)^{\nu-1}$, and show that equality holds only when G is a tree.

FEUE WHEP

8.4.4 Show that if G is a cycle of length n, then $\pi_k(G)=(k-1)^n+(-1)^n(k-1)$.

8.4.5 (*a*) Show that $\pi_k(G\vee K_1)=k\pi_{k-1}(G)$.

(*b*) Using (*a*) and exercise 8.4.4, show that if G is a wheel with n spokes, then $\pi_k(G)=k(k-2)^n+(-1)^n k(k-2)$.

8.4.6 Show that if $G_1, G_2, \ldots, G_\omega$ are the components of G, then $\pi_k(G)=\pi_k(G_1)\pi_k(G_2)\ldots\pi_k(G_\omega)$.

8.4.7 Show that if $G\cap H$ is complete, then $\pi_k(G\cup H)\pi_k(G\cap H)=\pi_k(G)\pi_k(H)$.

8.4.8* Show that no real root of $\pi_k(G)$ is greater than ν. (L. Lovász)

8.5 GIRTH AND CHROMATIC NUMBER

In any colouring of a graph, the vertices in a clique must all be assigned different colours. Thus a graph with a large clique necessarily has a high chromatic number. What is perhaps surprising is that there exist triangle-free graphs with arbitrarily high chromatic number. A recursive construction for such graphs was first described by Blanches Descartes (1954). (Her method, in fact, yields graphs that possess no cycles of length less than six.) We describe here an easier construction due to Mycielski (1955).

Theorem 8.7 For any positive integer k, there exists a k-chromatic graph containing no triangle.

Proof For $k=1$ and $k=2$, the graphs K_1 and K_2 have the required property. We proceed by induction on k. Suppose that we have already constructed a triangle-free graph G_k with chromatic number $k\geq 2$. Let the vertices of G_k be $v_1, v_2, \ldots, v_n$. Form a new graph G_{k+1} from G_k as follows: add $n+1$ new vertices $u_1, u_2, \ldots, u_n, v$, and then, for $1\leq i\leq n$, join u_i to the neighbours of v_i and to v. For example, if G_2 is K_2 then G_3 is the 5-cycle and G_4 the Grötzsch graph (see figure 8.10).

The graph G_{k+1} clearly has no triangles. For, since $\{u_1, u_2, \ldots, u_n\}$ is an independent set in G_{k+1}, no triangles can contain more than one u_i; and if $u_iv_jv_ku_i$ were a triangle in G_{k+1}, then $v_iv_jv_kv_i$ would be a triangle in G_k, contrary to assumption.

We now show that G_{k+1} is $(k+1)$-chromatic. Note, first, that G_{k+1} is certainly $(k+1)$-colourable, since any k-colouring of G_k can be extended to a $(k+1)$-colouring of G_{k+1} by colouring u_i the same as v_i, $1\leq i\leq n$, and then assigning a new colour to v. Therefore it remains to show that G_{k+1} is not k-colourable. If possible, consider a k-colouring of G_{k+1} in which, without loss of generality, v is assigned colour k. Clearly, no u_i can also have colour k. Now recolour each vertex v_i of colour k with the colour assigned to u_i.

FEUE WHEP

5709

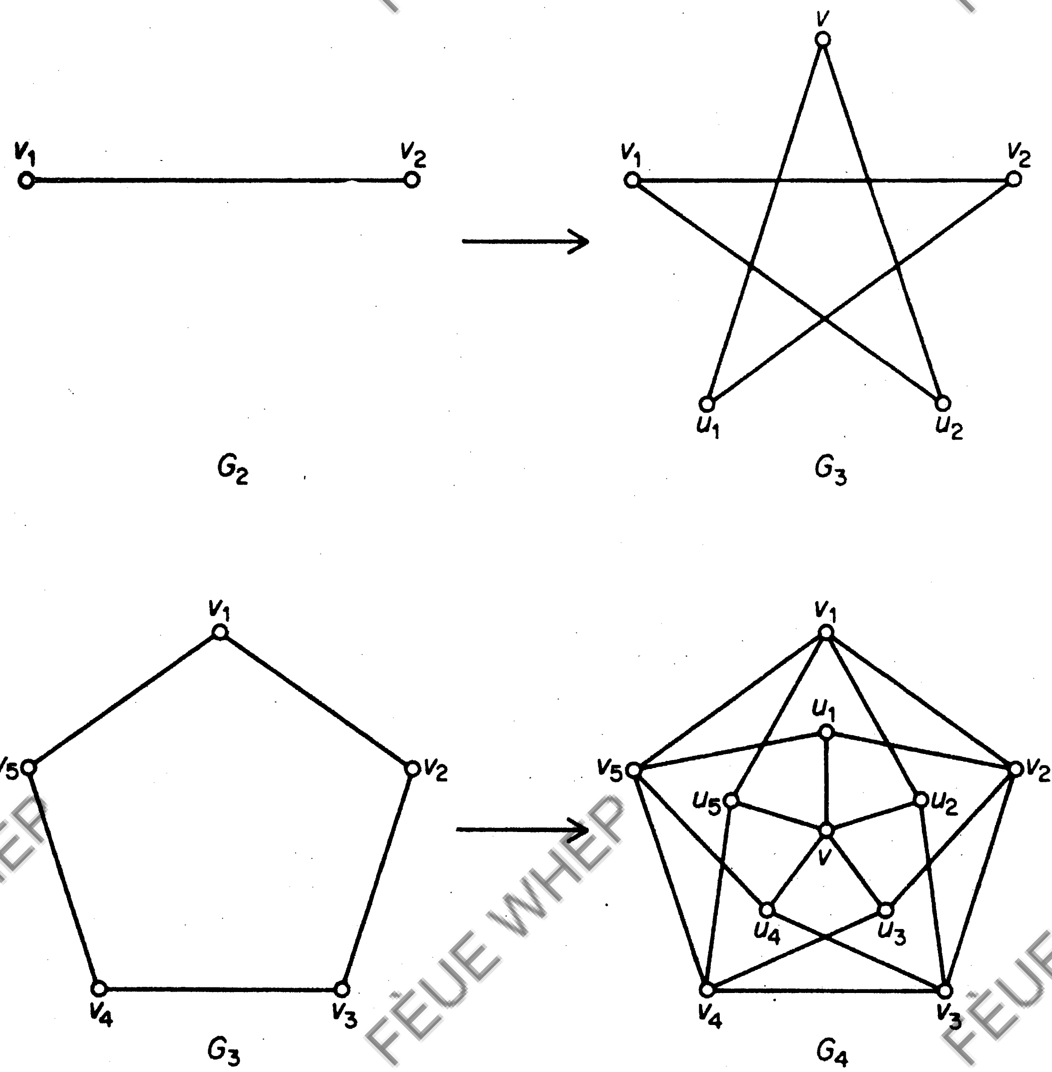


Figure 8.10. Mycielski's construction

This results in a $(k-1)$-colouring of the k-chromatic graph G_k. Therefore G_{k+1} is indeed $(k+1)$-chromatic. The theorem follows from the principle of induction □

By starting with the 2-chromatic graph K_2, the above construction yields, for all $k \geq 2$, a triangle-free k-chromatic graph on $3.2^{k-2}-1$ vertices.

We have already noted that there are graphs with girth six and arbitrary chromatic number. Using the probabilistic method, Erdös (1961) has, in fact, shown that, given any two integers $k \geq 2$ and $l \geq 2$, there is a graph with girth k and chromatic number l. Unfortunately, this application of the probabilistic method is not quite as straightforward as the one given in section 7.2, and we therefore choose to omit it. A constructive proof of Erdös' result has been given by Lovász (1968).

Exercises

8.5.1 Let $G_3, G_4, \ldots$ be the graphs obtained from $G_2 = K_2$, using Mycielski's construction. Show that each G_k is k-critical.

5710

8.5.2 (*a*)* Let G be a k-chromatic graph of girth at least six ($k \geq 2$). Form a new graph H as follows: Take $\binom{k\nu}{\nu}$ disjoint copies of G and a set S of $k\nu$ new vertices, and set up a one–one correspondence between the copies of G and the ν-element subsets of S. For each copy of G, join its vertices to the members of the corresponding ν-element subset of S by a matching. Show that H has chromatic number at least $k+1$ and girth at least six.

(*b*) Deduce that, for any $k \geq 2$, there exists a k-chromatic graph of girth six. (B. Descartes)

APPLICATIONS

8.6 A STORAGE PROBLEM

A company manufactures n chemicals $C_1, C_2, \ldots, C_n$. Certain pairs of these chemicals are incompatible and would cause explosions if brought into contact with each other. As a precautionary measure the company wishes to partition its warehouse into compartments, and store incompatible chemicals in different compartments. What is the least number of compartments into which the warehouse should be partitioned?

We obtain a graph G on the vertex set $\{v_1, v_2, \ldots, v_n\}$ by joining two vertices v_i and v_j if and only if the chemicals C_i and C_j are incompatible. It is easy to see that the least number of compartments into which the warehouse should be partitioned is equal to the chromatic number of G.

The solution of many problems of practical interest (of which the storage problem is one instance) involves finding the chromatic number of a graph. Unfortunately, no good algorithm is known for determining the chromatic number. Here we describe a systematic procedure which is basically 'enumerative' in nature. It is not very efficient for large graphs.

Since the chromatic number of a graph is the least number of independent sets into which its vertex set can be partitioned, we begin by describing a method for listing all the independent sets in a graph. Because every independent set is a subset of a maximal independent set, it suffices to determine all the maximal independent sets. In fact, our procedure first determines complements of maximal independent sets, that is, minimal coverings.

Observe that a subset K of V is a minimal covering of G if and only if, for each vertex v, either v belongs to K or all the neighbours of v belong to K (but not both). This provides us with a procedure for finding minimal coverings:

FOR EACH VERTEX v, CHOOSE EITHER v, OR ALL THE NEIGHBOURS OF v (8.2)

5711

To implement this procedure effectively, we make use of an algebraic device. First, we denote the instruction 'choose vertex v' simply by the symbol v. Then, given two instructions X and Y, the instructions 'either X or Y' and 'both X and Y' are denoted by $X+Y$ (the *logical sum*) and XY (the *logical product*), respectively. For example, the instruction 'choose either u and v or v and w' is written $uv+vw$. Formally, the logical sum and logical product behave like $\cup$ and $\cap$ for sets, and the algebraic laws that hold with respect to $\cup$ and $\cap$ also hold with respect to these two operations (see exercise 8.6.1). By using these laws, we can often simplify logical expressions; thus

$$\begin{aligned}(uv+vw)(u+vx) &= uvu+uvvx+vwu+vwvx\\ &= uv+uvx+vwu+vwx\\ &= uv+vwx\end{aligned}$$

Consider, now, the graph G of figure 8.11. Our prescription (8.2) for finding the minimal coverings in G is

$$(a+bd)(b+aceg)(c+bdef)(d+aceg)(e+bcdf)(f+ceg)(g+bdf) \quad (8.3)$$

It can be checked (exercise 8.6.2) that, on simplification, (8.3) reduces to

$$aceg+bcdeg+bdef+bcdf$$

In other words, 'choose a, c, e and g or b, c, d, e and g or b, d, e and f or b, c, d and f'. Thus $\{a, c, e, g\}$, $\{b, c, d, e, g\}$, $\{b, d, e, f\}$ and $\{b, c, d, f\}$ are the minimal coverings of G. On complementation, we obtain the list of all maximal independent sets of G: $\{b, d, f\}$, $\{a, f\}$, $\{a, c, g\}$ and $\{a, e, g\}$.

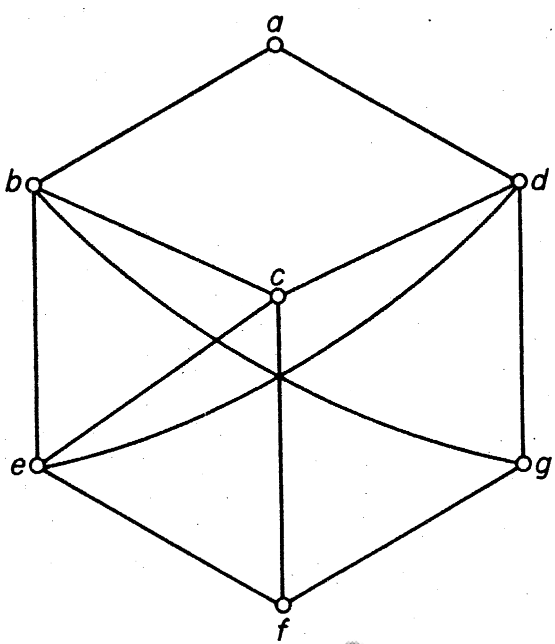


Figure 8.11

FEUE WHEP

5712

Now let us return to the problem of determining the chromatic number of a graph. A k-colouring $(V_1, V_2, \ldots, V_k)$ of G is said to be *canonical* if V_1 is a maximal independent set of G, V_2 is a maximal independent set of $G - V_1$, V_3 is a maximal independent set of $G - (V_1 \cup V_2)$, and so on. It is easy to see (exercise 8.6.3) that if G is k-colourable, then there exists a canonical k-colouring of G. By repeatedly using the above method for finding maximal independent sets, one can determine all the canonical colourings of G. The least number of colours used in such a colouring is then the chromatic number of G. For the graph G of figure 8.11, $\chi = 3$; a corresponding canonical colouring is $(\{b, d, f\}, \{a, e, g\}, \{c\})$.

Christofides (1971) gives some improvements on this procedure.

Exercises

8.6.1 Verify the associative, commutative, distributive and absorption laws for the logical sum and logical product.

8.6.2 Reduce (8.3) to $aceg + bcdeg + bdef + bcdf$.

8.6.3 Show that if G is k-vertex-colourable, then G has a canonical k-vertex colouring.

REFERENCES

Birkhoff, G. D. (1912). A determinant formula for the number of ways of coloring a map. *Ann. of Math.*, **14,** 42–46

Brooks, R. L. (1941). On colouring the nodes of a network. *Proc. Cambridge Philos. Soc.*, **37,** 194–97

Christofides, N. (1971). An algorithm for the chromatic number of a graph. *The Computer Journal*, **14,** 38–39

Descartes, B. (1954). Solution to advanced problem no. 4526. *Amer. Math. Monthly* **61,** 352

Dirac, G. A. (1952). A property of 4-chromatic graphs and some remarks on critical graphs. *J. London Math. Soc.*, **27,** 85–92

Dirac, G. A. (1953). The structure of k-chromatic graphs. *Fund. Math.*, **40,** 42–55

Erdös, P. (1961). Graph theory and probability II. *Canad. J. Math.*, **13,** 346–52

Grötzsch, H. (1958). Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg. Math.-Nat. Reihe*, **8,** 109–19

Hadwiger, H. (1943). Über eine Klassifikation der Streckenkomplexe. *Vierteljschr. Naturforsch. Gesellsch. Zürich*, **88,** 133–42

Hajós, G. (1961). Über eine Konstruktion nicht n-färbbarer Graphen. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg. Math.-Nat. Reihe.*, **10,** 116–17

5713

Lovász, L. (1968). On chromatic number of finite set systems. *Acta Math. Acad. Sci. Hungar.*, **19,** 59–67

Lovász, L. (1975). Three short proofs in graph theory, *J. Combinatorial Theory B*, **19,** 111–13.

Mycielski, J. (1955). Sur le coloriage des graphs. *Colloq. Math.*, **3**, 161–62

Read, R. C. (1968). An introduction to chromatic polynomials. *J. Combinatorial Theory*, **4,** 52–71

Tutte, W. T. (1970). On chromatic polynomials and the golden ratio. *J. Combinatorial Theory*, **9,** 289–96

Wagner, K. (1964). Beweis einer Abschwächung der Hadwiger-Vermutung. *Math. Ann.*, **153,** 139–41

FEUE WHEP

5714

9 Planar Graphs

9.1 PLANE AND PLANAR GRAPHS

A graph is said to be *embeddable in the plane*, or *planar*, if it can be drawn in the plane so that its edges intersect only at their ends. Such a drawing of a planar graph G is called a *planar embedding* of G. A planar embedding $\tilde{G}$ of G can itself be regarded as a graph isomorphic to G; the vertex set of $\tilde{G}$ is the set of points representing vertices of G, the edge set of $\tilde{G}$ is the set of lines representing edges of G, and a vertex of $\tilde{G}$ is incident with all the edges of $\tilde{G}$ that contain it. We therefore sometimes refer to a planar embedding of a planar graph as a *plane graph*. Figure 9.1*b* shows a planar embedding of the planar graph in figure 9.1*a*.

It is clear from the above definition that the study of planar graphs necessarily involves the topology of the plane. However, we shall not attempt here to be strictly rigorous in topological matters, and will be content to adopt a naive point of view toward them. This is done so as not to obscure the combinatorial aspect of the theory, which is our main interest.

The results of topology that are especially relevant in the study of planar graphs are those which deal with Jordan curves. (A *Jordan curve* is a continuous non-self-intersecting curve whose origin and terminus coincide.) The union of the edges in a cycle of a plane graph constitutes a Jordan curve; this is the reason why properties of Jordan curves come into play in planar graph theory. We shall recall a well-known theorem about Jordan curves and use it to demonstrate the nonplanarity of K_5.

Let J be a Jordan curve in the plane. Then the rest of the plane is partitioned into two disjoint open sets called the *interior* and *exterior* of J. We shall denote the interior and exterior of J, respectively, by $\text{int}\, J$ and $\text{ext}\, J$, and their closures by $\text{Int}\, J$ and $\text{Ext}\, J$. Clearly $\text{Int}\, J \cap \text{Ext}\, J = J$. The *Jordan curve theorem* states that any line joining a point in $\text{int}\, J$ to a point in $\text{ext}\, J$ must meet J in some point (see figure 9.2). Although this theorem is intuitively obvious, a formal proof of it is quite difficult.

Theorem 9.1 K_5 is nonplanar.

Proof By contradiction. If possible let G be a plane graph corresponding to K_5. Denote the vertices of G by v_1, v_2, v_3, v_4 and v_5. Since G is complete, any two of its vertices are joined by an edge. Now the cycle $C = v_1v_2v_3v_1$ is a Jordan curve in the plane, and the point v_4 must lie either in $\text{int}\, C$ or $\text{ext}\, C$.

FEUE WHEP

5715

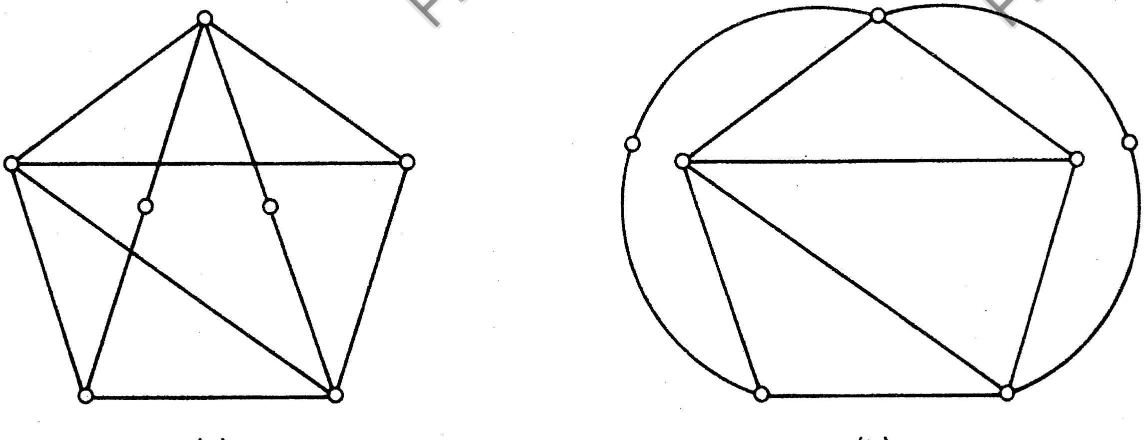


Figure 9.1. (a) A planar graph G; (b) a planar embedding of G

We shall suppose that $v_4 \in \text{int } C$. (The case where $v_4 \in \text{ext } C$ can be dealt with in a similar manner.) Then the edges v_4v_1, v_4v_2 and v_4v_3 divide int C into the three regions int C_1, int C_2 and int C_3, where $C_1 = v_1v_4v_2v_1$, $C_2 = v_2v_4v_3v_2$ and $C_3 = v_3v_4v_1v_3$ (see figure 9.3).

Now v_5 must lie in one of the four regions ext C, int C_1, int C_2 and int C_3. If $v_5 \in \text{ext } C$ then, since $v_4 \in \text{int } C$, it follows from the Jordan curve theorem that the edge v_4v_5 must meet C in some point. But this contradicts the assumption that G is a plane graph. The cases $v_5 \in \text{int } C_i$, $i = 1, 2, 3$, can be disposed of in like manner □

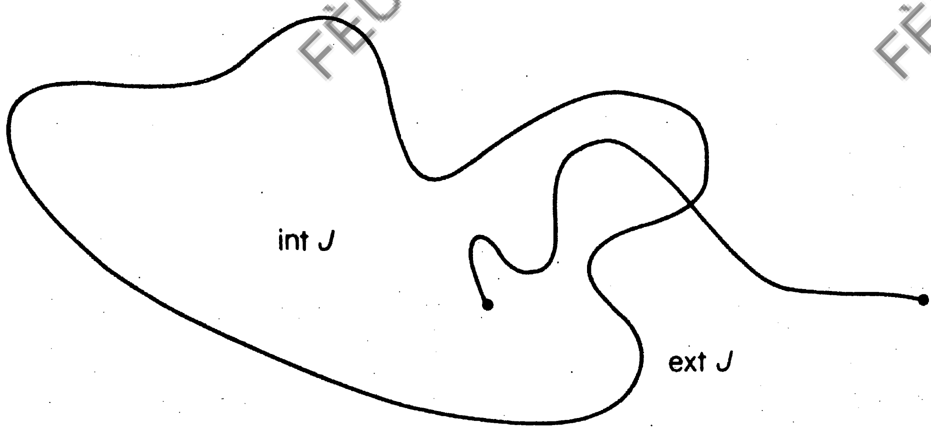


Figure 9.2

A similar argument can be used to establish that $K_{3,3}$, too, is nonplanar (exercise 9.1.1). We shall see in section 9.5 that, on the other hand, every nonplanar graph contains a subdivision of either K_5 or $K_{3,3}$.

The notion of a planar embedding extends to other surfaces.† A graph G is said to be *embeddable* on a surface S if it can be drawn in S so that its

† A *surface* is a 2-dimensional manifold. Closed surfaces are divided into two classes, orientable and non-orientable. The sphere and the torus are examples of orientable surfaces; the projective plane and the Möbius band are non-orientable. For a detailed account of embeddings of graphs on surfaces the reader is referred to Fréchet and Fan (1967).

5716

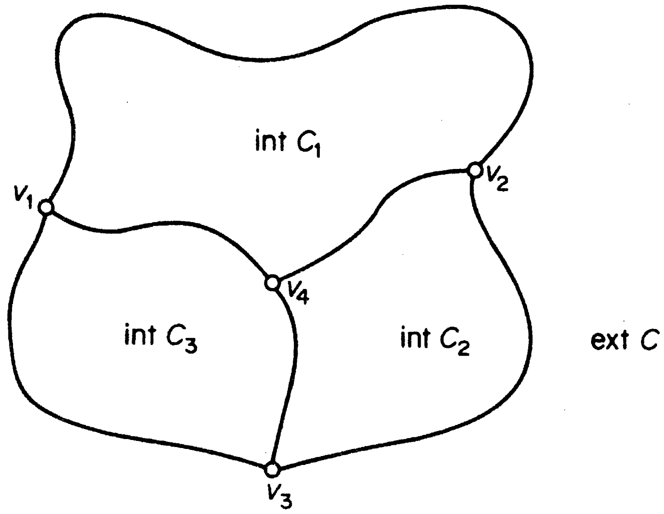


Figure 9.3

edges intersect only at their ends; such a drawing (if one exists) is called an *embedding* of G on S. Figure 9.4*a* shows an embedding of K_5 on the torus, and figure 9.4*b* an embedding of $K_{3,3}$ on the Möbius band. The torus is represented as a rectangle in which opposite sides are identified, and the Möbius band as a rectangle whose two ends are identified after one half-twist.

We have seen that not all graphs can be embedded in the plane; this is also true of other surfaces. It can be shown (see, for example, Fréchet and Fan, 1967) that, for every surface S, there exist graphs which are not embeddable on S. Every graph can, however, be 'embedded' in 3-dimensional space $\mathscr{R}^3$ (exercise 9.1.3).

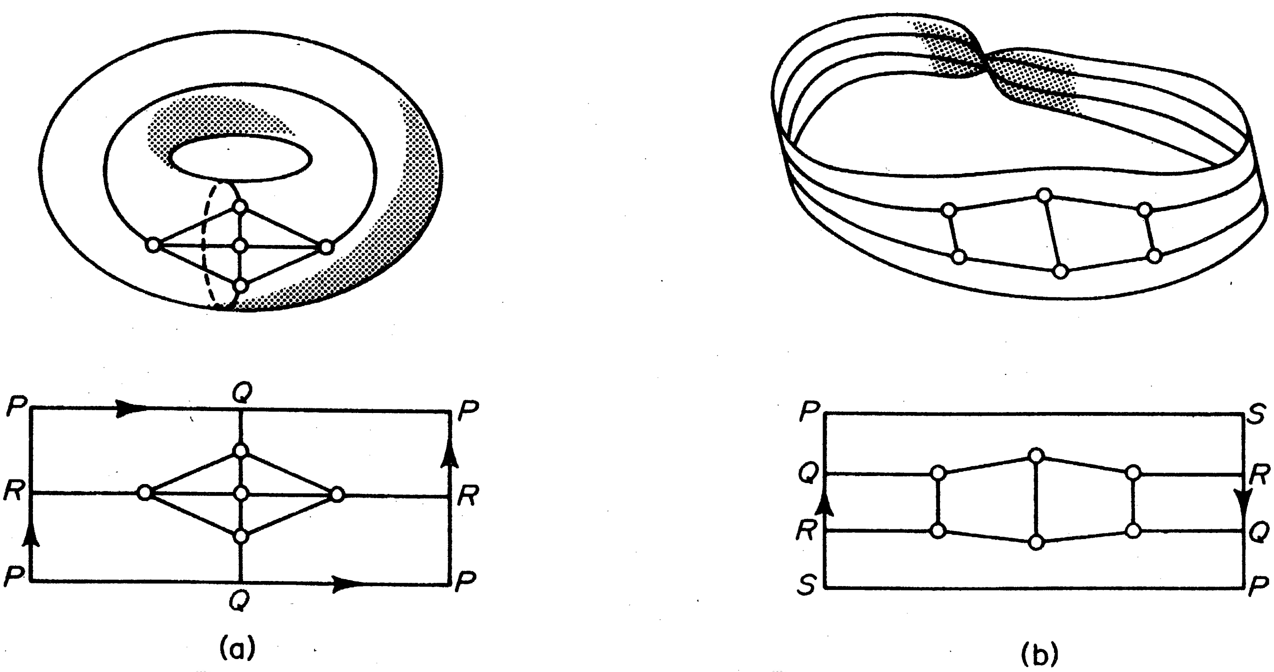


Figure 9.4. (*a*) An embedding of K_5 on the torus; (*b*) an embedding of $K_{3,3}$ on the Möbius band

FEUE WHEP

5717

Planar graphs and graphs embeddable on the sphere are one and the same. To show this we make use of a mapping known as stereographic projection. Consider a sphere S resting on a plane P, and denote by z the point of S that is diagonally opposite the point of contact of S and P. The mapping $\pi : S\backslash\{z\} \to P$, defined by $\pi(s) = p$ if and only if the points z, s and p are collinear, is called *stereographic projection* from z; it is illustrated in figure 9.5.

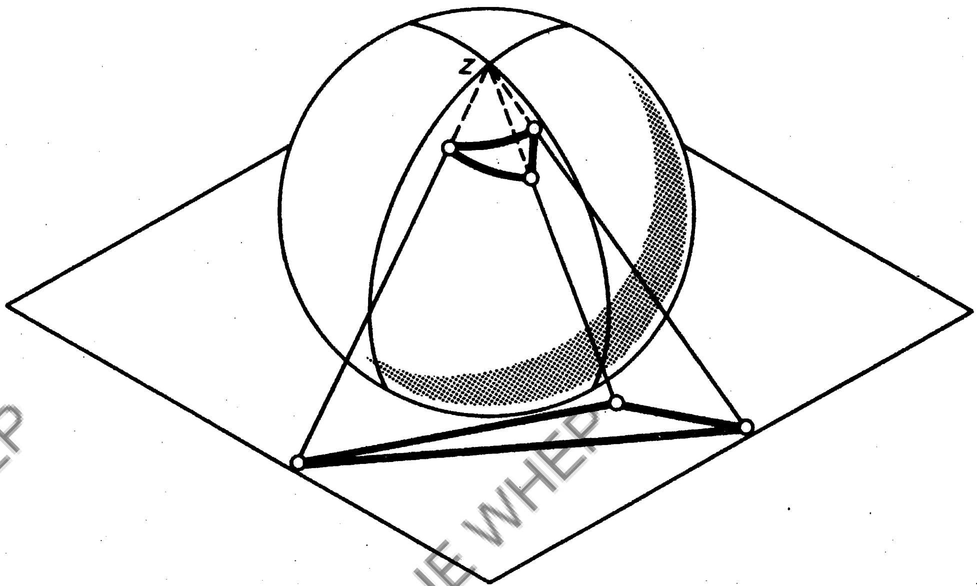


Figure 9.5. Stereographic projection

Theorem 9.2 A graph G is embeddable in the plane if and only if it is embeddable on the sphere.

Proof Suppose G has an embedding $\tilde{G}$ on the sphere. Choose a point z of the sphere not in $\tilde{G}$. Then the image of $\tilde{G}$ under stereographic projection from z is an embedding of G in the plane. The converse is proved similarly □

On many occasions it is advantageous to consider embeddings of planar graphs on the sphere; one instance is provided by the proof of theorem 9.3 in the next section.

Exercises

9.1.1 Show that $K_{3,3}$ is nonplanar.

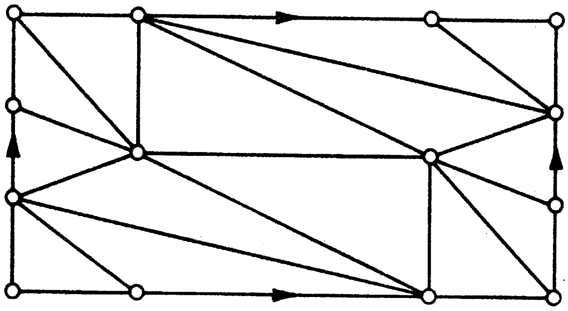
9.1.2 (*a*) Show that $K_5 - e$ is planar for any edge e of K_5.
(*b*) Show that $K_{3,3} - e$ is planar for any edge e of $K_{3,3}$.

9.1.3 Show that all graphs are 'embeddable' in $\mathcal{R}^3$.

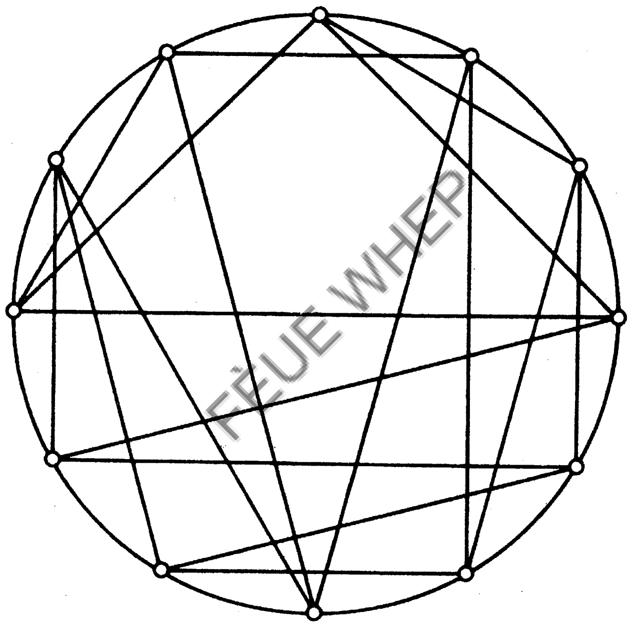
FEUE WHEP

5718

9.1.4 Verify that the following is an embedding of K_7 on the torus:



9.1.5 Find a planar embedding of the following graph in which each edge is a straight line.
(Fáry, 1948 has proved that every simple planar graph has such an embedding.)



9.2 DUAL GRAPHS

A plane graph G partitions the rest of the plane into a number of connected regions; the closures of these regions are called the *faces* of G. Figure 9.6 shows a plane graph with six faces, f_1, f_2, f_3, f_4, f_5 and f_6. The notion of a face applies also to embeddings of graphs on other surfaces. We shall denote by $F(G)$ and $\phi(G)$, respectively, the set of faces and the number of faces of a plane graph G.

Each plane graph has exactly one unbounded face, called the *exterior face;* in the plane graph of figure 9.6, f_1 is the exterior face.

Theorem 9.3 Let v be a vertex of a planar graph G. Then G can be embedded in the plane in such a way that v is on the exterior face of the embedding.

FÊUE WHEP

5719

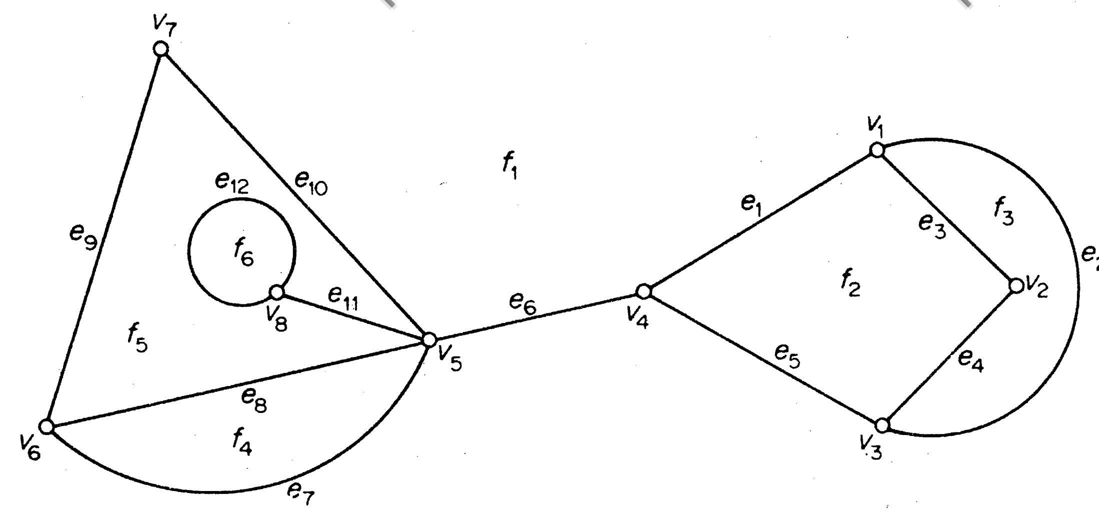


Figure 9.6. A plane graph with six faces

Proof Consider an embedding $\tilde{G}$ of G on the sphere; such an embedding exists by virtue of theorem 9.2. Let z be a point in the interior of some face containing v, and let $\pi(\tilde{G})$ be the image of $\tilde{G}$ under stereographic projection from z. Clearly $\pi(\tilde{G})$ is a planar embedding of G of the desired type. □

We denote the boundary of a face f of a plane graph G by $b(f)$. If G is connected, then $b(f)$ can be regarded as a closed walk in which each cut edge of G in $b(f)$ is traversed twice; when $b(f)$ contains no cut edges, it is a cycle of G. For example, in the plane graph of figure 9.6,

$$b(f_2) = v_1e_3v_2e_4v_3e_5v_4e_1v_1$$

and

$$b(f_5) = v_7e_{10}v_5e_{11}v_8e_{12}v_8e_{11}v_5e_8v_6e_9v_7$$

A face f is said to be *incident* with the vertices and edges in its boundary. If e is a cut edge in a plane graph, just one face is incident with e; otherwise, there are two faces incident with e. We say that an edge *separates* the faces incident with it. The *degree*, $d_G(f)$, of a face f is the number of edges with which it is incident (that is, the number of edges in $b(f)$), cut edges being counted twice. In figure 9.6, f_1 is incident with the vertices v_1, v_3, v_4, v_5, v_6, v_7 and the edges e_1, e_2, e_5, e_6, e_7, e_9, e_{10}; e_1 separates f_1 from f_2 and e_{11} separates f_5 from f_5; $d(f_2) = 4$ and $d(f_5) = 6$.

Given a plane graph G, one can define another graph G^* as follows: corresponding to each face f of G there is a vertex f^* of G^*, and corresponding to each edge e of G there is an edge e^* of G^*; two vertices f^* and g^* are joined by the edge e^* in G^* if and only if their corresponding faces f and g are separated by the edge e in G. The graph G^* is called the *dual* of G. A plane graph and its dual are shown in figures 9.7a and 9.7b.

It is easy to see that the dual G^* of a plane graph G is planar; in fact,

5720

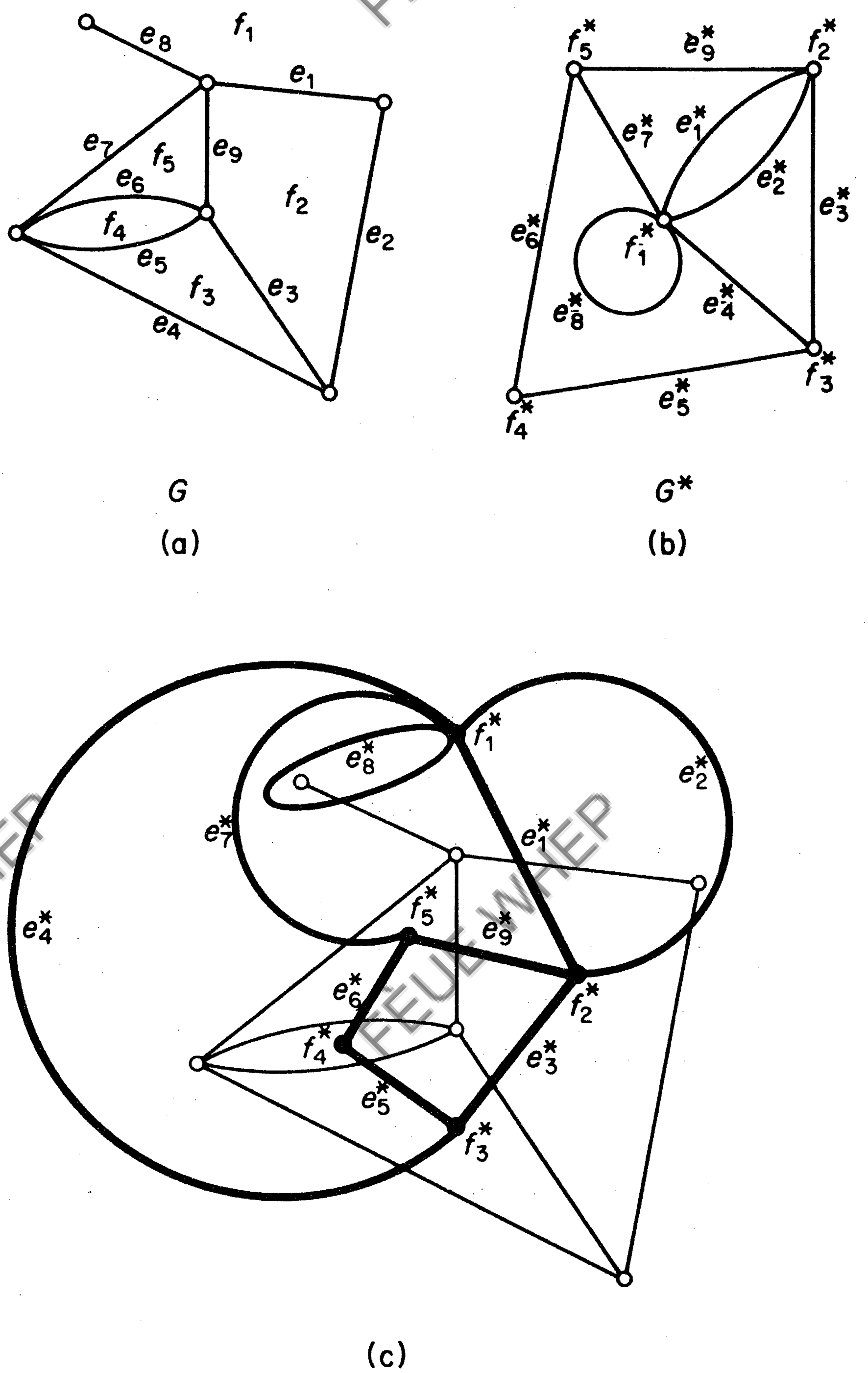


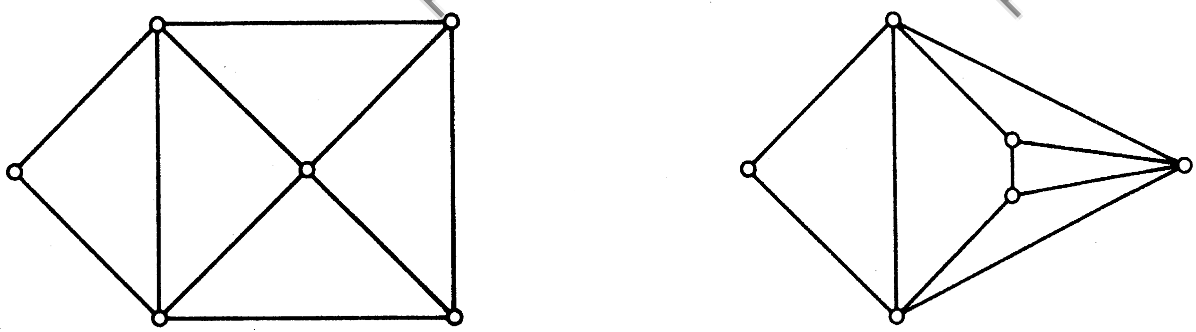
Figure 9.7. A plane graph and its dual

there is a natural way to embed G^* in the plane. We place each vertex f^* in the corresponding face f of G, and then draw each edge e^* in such a way that it crosses the corresponding edge e of G exactly once (and crosses no other edge of G). This procedure is illustrated in figure 9.7*c*, where the dual is indicated by heavy points and lines. It is intuitively clear that we can always draw the dual as a plane graph in this way, but we shall not prove this fact. Note that if e is a loop of G, then e^* is a cut edge of G^*, and vice versa.

Although defined abstractly, it is sometimes convenient to regard the dual

FEUE WHEP

5721



(a) (b)

Figure 9.8. Isomorphic plane graphs with nonisomorphic duals

G^* of a plane graph G as a plane graph (embedded as described above). One can then consider the dual G^{**} of G^*, and it is not difficult to prove that, when G is connected, $G^{**} \cong G$ (exercise 9.2.4); a glance at figure 9.7c will indicate why this is so.

It should be noted that isomorphic plane graphs may have nonisomorphic duals. For example, the plane graphs in figure 9.8 are isomorphic, but their duals are not—the plane graph of figure 9.8*a* has a face of degree five, whereas the plane graph of figure 9.8*b* has no such face. Thus the notion of a dual is meaningful only for plane graphs, and cannot be extended to planar graphs in general.

The following relations are direct consequences of the definition of G^*:

$$\begin{aligned} \nu(G^*) &= \phi(G) \\ \varepsilon(G^*) &= \varepsilon(G) \\ d_{G^*}(f^*) &= d_G(f) \quad \text{for all} \quad f \in F(G) \end{aligned} \tag{9.1}$$

Theorem 9.4 If G is a plane graph, then

$$\sum_{f \in F} d(f) = 2\varepsilon$$

Proof Let G^* be the dual of G. Then

$$\begin{aligned} \sum_{f \in F(G)} d(f) &= \sum_{f^* \in V(G^*)} d(f^*) && \text{by (9.1)} \\ &= 2\varepsilon(G^*) && \text{by theorem 1.1} \\ &= 2\varepsilon(G) && \text{by (9.1)} \quad \square \end{aligned}$$

Exercises

9.2.1 (*a*) Show that a graph is planar if and only if each of its blocks is planar.
(*b*) Deduce that a minimal nonplanar graph is a simple block.

9.2.2 A plane graph is *self-dual* if it is isomorphic to its dual.
(*a*) Show that if G is self-dual, then $\varepsilon = 2\nu - 2$.
(*b*) For each $n \geq 4$, find a self-dual plane graph on n vertices.

FEUE WHEP

5722

9.2.3 (*a*) Show that B is a bond of a plane graph G if and only if $\{e^* \in E(G^*) \mid e \in B\}$ is a cycle of G^*.
(*b*) Deduce that the dual of an eulerian plane graph is bipartite.

9.2.4 Let G be a plane graph. Show that
(*a*) $G^{**} \cong G$ if and only if G is connected;
(*b*) $\chi(G^{**}) = \chi(G)$.

9.2.5 Let T be a spanning tree of a connected plane graph G, and let $E^* = \{e^* \in E(G^*) \mid e \notin E(T)\}$. Show that $T^* = G^*[E^*]$ is a spanning tree of G^*.

9.2.6 A *plane triangulation* is a plane graph in which each face has degree three. Show that every simple plane graph is a spanning subgraph of some simple plane triangulation ($\nu \geq 3$).

9.2.7 Let G be a simple plane triangulation with $\nu \geq 4$. Show that G^* is a simple 2-edge-connected 3-regular planar graph.

9.2.8* Show that any plane triangulation G contains a bipartite subgraph with $2\varepsilon(G)/3$ edges. (F. Harary, D. Matula)

9.3 EULER'S FORMULA

There is a simple formula relating the numbers of vertices, edges and faces in a connected plane graph. It is known as *Euler's formula* because Euler established it for those plane graphs defined by the vertices and edges of polyhedra.

Theorem 9.5 If G is a connected plane graph, then

$$\nu - \varepsilon + \phi = 2$$

Proof By induction on ϕ, the number of faces of G. If $\phi = 1$, then each edge of G is a cut edge and so G, being connected, is a tree. In this case $\varepsilon = \nu - 1$, by theorem 2.2, and the theorem clearly holds. Suppose that it is true for all connected plane graphs with fewer than n faces, and let G be a connected plane graph with $n \geq 2$ faces. Choose an edge e of G that is not a cut edge. Then $G - e$ is a connected plane graph and has $n - 1$ faces, since the two faces of G separated by e combine to form one face of $G - e$. By the induction hypothesis

$$\nu(G-e) - \varepsilon(G-e) + \phi(G-e) = 2$$

and, using the relations

$$\nu(G-e) = \nu(G) \qquad \varepsilon(G-e) = \varepsilon(G) - 1 \qquad \phi(G-e) = \phi(G) - 1$$

we obtain

$$\nu(G) - \varepsilon(G) + \phi(G) = 2$$

The theorem follows by the principle of induction □

FEUE WHEP

5723

Corollary 9.5.1 All planar embeddings of a given connected planar graph have the same number of faces.

Proof Let G and H be two planar embeddings of a given connected planar graph. Since $G \cong H$, $\nu(G) = \nu(H)$ and $\varepsilon(G) = \varepsilon(H)$. Applying theorem 9.5, we have

$$\phi(G) = \varepsilon(G) - \nu(G) + 2 = \varepsilon(H) - \nu(H) + 2 = \phi(H) \quad \square$$

Corollary 9.5.2 If G is a simple planar graph with $\nu \geq 3$, then $\varepsilon \leq 3\nu - 6$.

Proof It clearly suffices to prove this for connected graphs. Let G be a simple connected graph with $\nu \geq 3$. Then $d(f) \geq 3$ for all $f \in F$, and

$$\sum_{f \in F} d(f) \geq 3\phi$$

By theorem 9.4

$$2\varepsilon \geq 3\phi$$

Thus, from theorem 9.5

$$\nu - \varepsilon + 2\varepsilon/3 \geq 2$$

or

$$\varepsilon \leq 3\nu - 6 \quad \square$$

Corollary 9.5.3 If G is a simple planar graph, then $\delta \leq 5$.

Proof This is trivial for $\nu = 1, 2$. If $\nu \geq 3$, then, by theorem 1.1 and corollary 9.5.2,

$$\delta\nu \leq \sum_{v \in V} d(v) = 2\varepsilon \leq 6\nu - 12$$

It follows that $\delta \leq 5$ $\square$

We have already seen that K_5 and $K_{3,3}$ are nonplanar (theorem 9.1 and exercise 9.1.1). Here, we shall derive these two results as corollaries of theorem 9.5.

Corollary 9.5.4 K_5 is nonplanar.

Proof If K_5 were planar then, by corollary 9.5.2, we would have

$$10 = \varepsilon(K_5) \leq 3\nu(K_5) - 6 = 9$$

Thus K_5 must be nonplanar $\square$

Corollary 9.5.5 $K_{3,3}$ is nonplanar.

Proof Suppose that $K_{3,3}$ is planar and let G be a planar embedding of $K_{3,3}$. Since $K_{3,3}$ has no cycles of length less than four, every face of G must

5724

have degree at least four. Therefore, by theorem 9.4, we have

$$4\phi \leq \sum_{f \in F} d(f) = 2\varepsilon = 18$$

That is

$$\phi \leq 4$$

Theorem 9.5 now implies that

$$2 = \nu - \varepsilon + \phi \leq 6 - 9 + 4 = 1$$

which is absurd □

Exercises

9.3.1 (*a*) Show that if G is a connected planar graph with girth $k \geq 3$, then $\varepsilon \leq k(\nu - 2)/(k - 2)$.
(*b*) Using (*a*), show that the Petersen graph is nonplanar.

9.3.2 Show that every planar graph is 6-vertex-colourable.

9.3.3 (*a*) Show that if G is a simple planar graph with $\nu \geq 11$, then G^c is nonplanar.
(*b*) Find a simple planar graph G with $\nu = 8$ such that G^c is also planar.

9.3.4 The *thickness* $\theta(G)$ of G is the minimum number of planar graphs whose union is G. (Thus $\theta(G) = 1$ if and only if G is planar.)
(*a*) Show that $\theta(G) \geq \{\varepsilon/(3\nu - 6)\}$.
(*b*) Deduce that $\theta(K_\nu) \geq \{\nu(\nu - 1)/6(\nu - 2)\}$ and show, using exercise 9.3.3*b*, that equality holds for all $\nu \leq 8$.

9.3.5 Use the result of exercise 9.2.5 to deduce Euler's formula.

9.3.6 Show that if G is a plane triangulation, then $\varepsilon = 3\nu - 6$.

9.3.7 Let $S = \{x_1, x_2, \ldots, x_n\}$ be a set of $n \geq 3$ points in the plane such that the distance between any two points is at least one. Show that there are at most $3n - 6$ pairs of points at distance exactly one.

9.4 BRIDGES

In the study of planar graphs, certain subgraphs, called bridges, play an important rôle. We shall discuss properties of these subgraphs in this section.

Let H be a given subgraph of a graph G. We define a relation $\sim$ on $E(G)\backslash E(H)$ by the condition that $e_1 \sim e_2$ if there exists a walk W such that

(i) the first and last edges of W are e_1 and e_2, respectively, and
(ii) W is internally-disjoint from H (that is, no internal vertex of W is a vertex of H).

It is easy to verify that $\sim$ is an equivalence relation on $E(G)\backslash E(H)$. A subgraph of $G - E(H)$ induced by an equivalence class under the relation $\sim$

FEUE WHEP

5725

is called a *bridge* of H in G. It follows immediately from the definition that if B is a bridge of H, then B is a connected graph and, moreover, that any two vertices of B are connected by a path that is internally-disjoint from H. It is also easy to see that two bridges of H have no vertices in common except, possibly, for vertices of H. For a bridge B of H, we write $V(B)\cap V(H)=V(B, H)$, and call the vertices in this set the *vertices of attachment* of B to H. Figure 9.9 shows a variety of bridges of a cycle in a graph; edges of different bridges are represented by different kinds of lines.

In this section we are concerned with the study of bridges of a cycle C. Thus, to avoid repetition, we shall abbreviate 'bridge of C' to 'bridge' in the coming discussion; all bridges will be understood to be bridges of a given cycle C.

In a connected graph every bridge has at least one vertex of attachment, and in a block every bridge has at least two vertices of attachment. A bridge with k vertices of attachment is called a *k-bridge*. Two k-bridges with the same vertices of attachment are *equivalent* k-bridges; for example, in figure 9.9, B_1 and B_2 are equivalent 3-bridges.

The vertices of attachment of a k-bridge B with $k\geq 2$ effect a partition of C into edge-disjoint paths, called the *segments* of B. Two bridges *avoid* one another if all the vertices of attachment of one bridge lie in a single segment of the other bridge; otherwise they *overlap*. In figure 9.9, B_2 and B_3 avoid one another, whereas B_1 and B_2 overlap. Two bridges B and B' are *skew* if there are four distinct vertices u, v, u' and v' of C such that u and v are vertices of attachment of B, u' and v' are vertices of attachment of B', and the four vertices appear in the cyclic order u, u', v, v' on C. In figure 9.9, B_3 and B_4 are skew, but B_1 and B_2 are not.

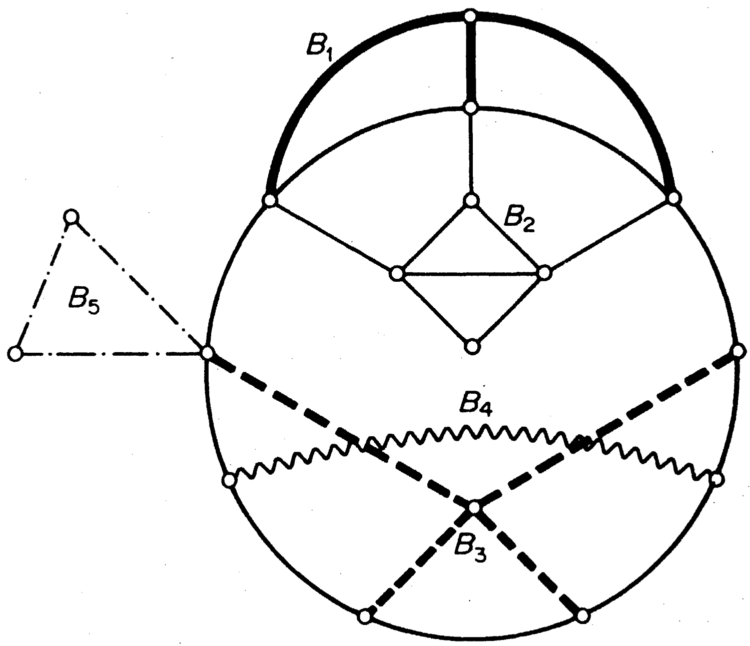


Figure 9.9. Bridges in a graph

Theorem 9.6 If two bridges overlap, then either they are skew or else they are equivalent 3-bridges.

Proof Suppose that the bridges B and B' overlap. Clearly, each must have at least two vertices of attachment. Now if either B or B' is a 2-bridge, it is easily verified that they must be skew. We may therefore assume that both B and B' have at least three vertices of attachment. There are two cases.

Case 1 B and B' are not equivalent bridges. Then B' has a vertex of attachment u' between two consecutive vertices of attachment u and v of B. Since B and B' overlap, some vertex of attachment v' of B' does not lie in the segment of B connecting u and v. It now follows that B and B' are skew.

Case 2 B and B' are equivalent k-bridges, $k \geq 3$. If $k \geq 4$, then B and B' are clearly skew; if $k = 3$, they are equivalent 3-bridges □

Theorem 9.7 If a bridge B has three vertices of attachment v_1, v_2 and v_3, then there exists a vertex v_0 in $V(B)\backslash V(C)$ and three paths P_1, P_2 and P_3 in B joining v_0 to v_1, v_2 and v_3, respectively, such that, for $i \neq j$, P_i and P_j have only the vertex v_0 in common (see figure 9.10).

Proof Let P be a (v_1, v_2)-path in B, internally-disjoint from C. P must have an internal vertex v, since otherwise the bridge B would be just P, and would not contain a third vertex v_3. Let Q be a (v_3, v)-path in B, internally-disjoint from C, and let v_0 be the first vertex of Q on P. Denote by P_1 the (v_0, v_1)-section of P^{-1}, by P_2 the (v_0, v_2)-section of P, and by P_3 the (v_0, v_3)-section of Q^{-1}. Clearly P_1, P_2 and P_3 satisfy the required conditions □

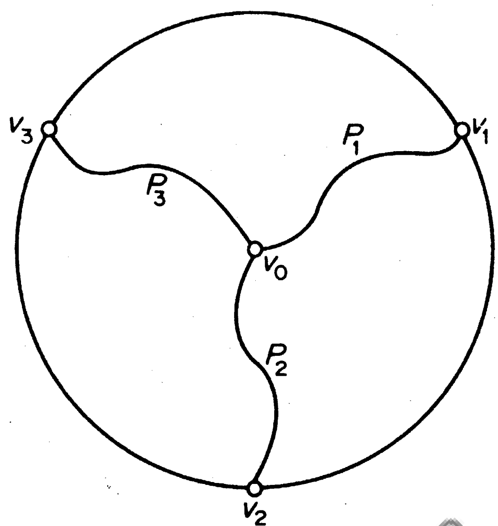


Figure 9.10

FEUE WHEP

5727

We shall now consider bridges in plane graphs. Suppose that G is a plane graph and that C is a cycle in G. Then C is a Jordan curve in the plane, and each edge of $E(G)\backslash E(C)$ is contained in one of the two regions Int C and Ext C. It follows that a bridge of C is contained entirely in Int C or Ext C. A bridge contained in Int C is called an *inner bridge*, and a bridge contained in Ext C, an *outer bridge*. In figure 9.11 B_1 and B_2 are inner bridges, and B_3 and B_4 are outer bridges.

Theorem 9.8 Inner (outer) bridges avoid one another.

Proof By contradiction. Let B and B' be two inner bridges that overlap. Then, by theorem 9.6, they must be either skew or equivalent 3-bridges.

Case 1 B and B' are skew. By definition, there exist distinct vertices u and v in B and u' and v' in B', appearing in the cyclic order u, u', v, v' on C. Let P be a (u, v)-path in B and P' a (u', v')-path in B', both internally-disjoint from C. The two paths P and P' cannot have an internal vertex in common because they belong to different bridges. At the same time, both P and P' must be contained in Int C because B and B' are inner bridges. By the Jordan curve theorem, G cannot be a plane graph, contrary to hypothesis (see figure 9.12).

Case 2 B and B' are equivalent 3-bridges. Let the common set of vertices of attachment be $\{v_1, v_2, v_3\}$. By theorem 9.7, there exist in B a vertex v_0 and three paths P_1, P_2 and P_3 joining v_0 to v_1, v_2 and v_3, respectively, such that, for $i \neq j$, P_i and P_j have only the vertex v_0 in common. Similarly, B' has a vertex v_0' and three paths P_1', P_2' and P_3' joining v_0' to v_1, v_2 and v_3, respectively, such that, for $i \neq j$, P_i' and P_j' have only the vertex v_0' in common (see figure 9.13).

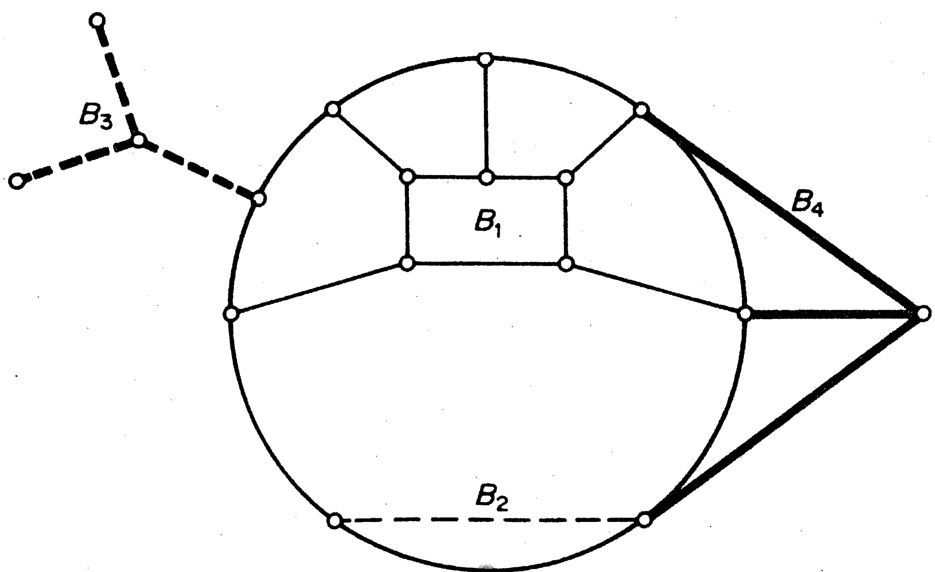


Figure 9.11. Bridges in a plane graph

FEUE WHEP

5728

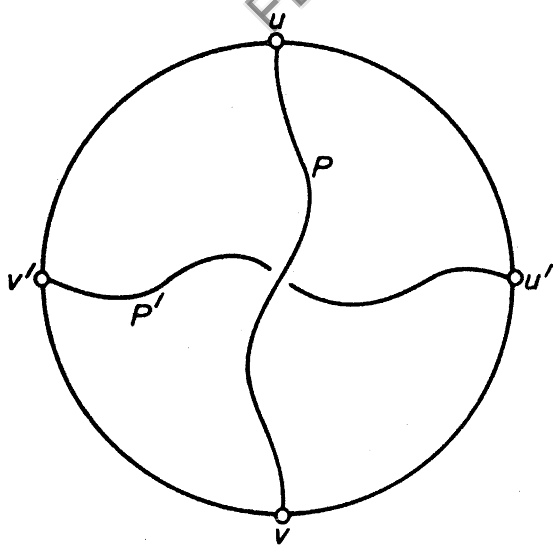


Figure 9.12

Now the paths P_1, P_2 and P_3 divide Int C into three regions, and v_0' must be in the interior of one of these regions. Since only two of the vertices v_1, v_2 and v_3 can lie on the boundary of the region containing v_0', we may assume, by symmetry, that v_3 is not on the boundary of this region. By the Jordan curve theorem, the path P_3' must cross either P_1, P_2 or C. But since B and B' are distinct inner bridges, this is clearly impossible.

We conclude that inner bridges avoid one another. Similarly, outer bridges avoid one another □

Let G be a plane graph. An inner bridge B of a cycle C in G is *transferable* if there exists a planar embedding $\tilde{G}$ of G which is identical to G itself, except that B is an outer bridge of C in $\tilde{G}$. The plane graph $\tilde{G}$ is said to be obtained from G by *transferring* B. Figure 9.14 illustrates the transfer of a bridge.

Theorem 9.9 An inner bridge that avoids every outer bridge is transferable.

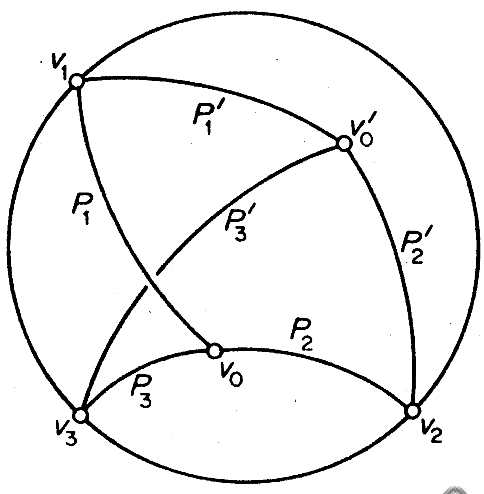


Figure 9.13

5729

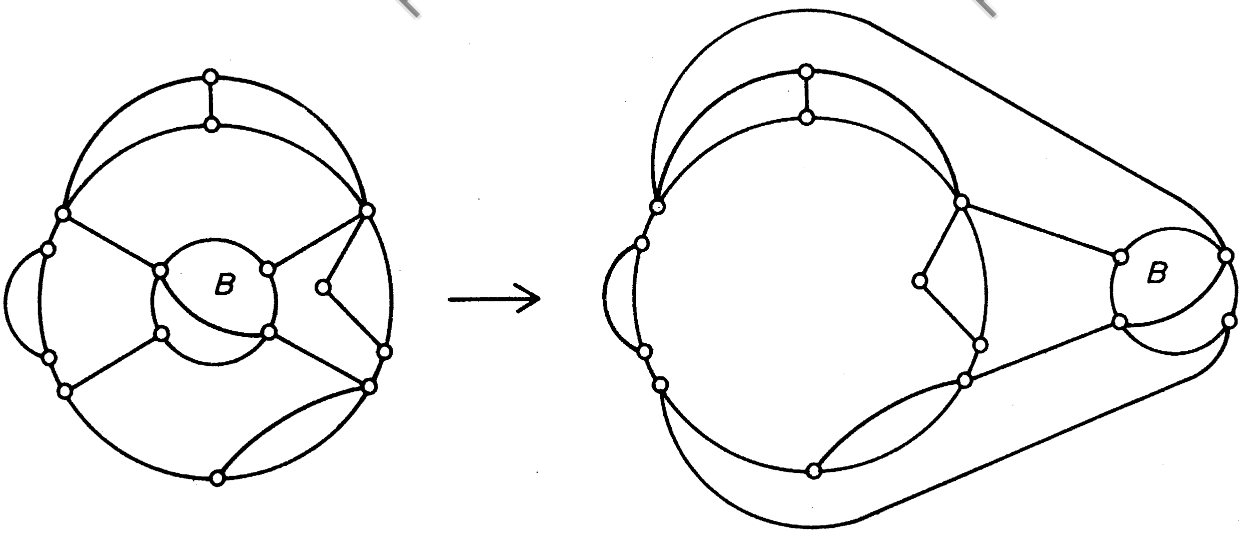


Figure 9.14. The transfer of a bridge

Proof Let B be an inner bridge that avoids every outer bridge. Then the vertices of attachment of B to C all lie on the boundary of some face of G contained in Ext C. B can now be drawn in this face, as shown in figure 9.15 □

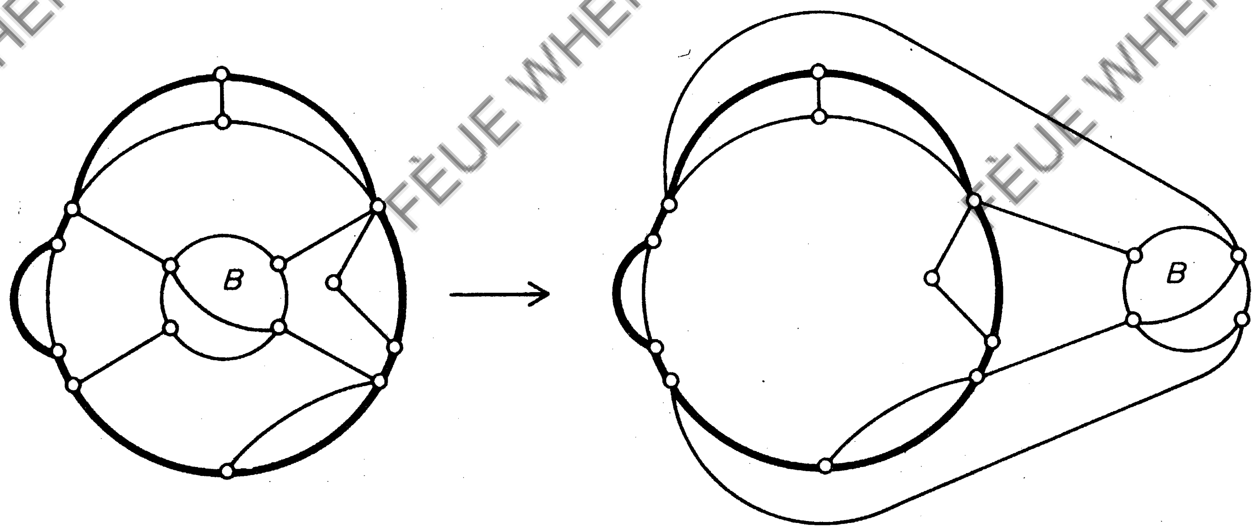


Figure 9.15

Theorem 9.9 is crucial to the proof of Kuratowski's theorem, which will be proved in the next section.

Exercises

9.4.1 Show that if B and B' are two distinct bridges, then $V(B)\cap V(B')\subseteq V(C)$.

9.4.2 Let u, x, v and y (in that cyclic order) be four distinct vertices of attachment of a bridge B to a cycle C in a plane graph. Show that there is a (u, v)-path P and an (x, y)-path Q in B such that (i) P and Q are internally-disjoint from C, and (ii) $|V(P)\cap V(Q)|\geq 1$.

FEUE WHEP

5730

9.4.3 (*a*) Let $C = v_1v_2 \ldots v_nv_1$ be a longest cycle in a nonhamiltonian connected graph G. Show that

(i) there exists a bridge B such that $V(B)\backslash V(C) \neq \emptyset$;

(ii) if v_i and v_j are vertices of attachment of B, then $v_{i+1}v_{j+1} \notin E$.

(*b*) Deduce that if $\alpha \leqslant \kappa$, then G is hamiltonian.

(V. Chvátal and P. Erdös)

9.5 KURATOWSKI'S THEOREM

Since planarity is such a fundamental property, it is clearly of importance to know which graphs are planar and which are not. We have already noted that, in particular, K_5 and $K_{3,3}$ are nonplanar and that any proper subgraph of either of these graphs is planar (exercise 9.1.2). A remarkably simple characterisation of planar graphs was given by Kuratowski (1930). This section is devoted to a proof of Kuratowski's theorem.

The following lemmas are simple observations, and we leave their proofs as an exercise (9.5.1).

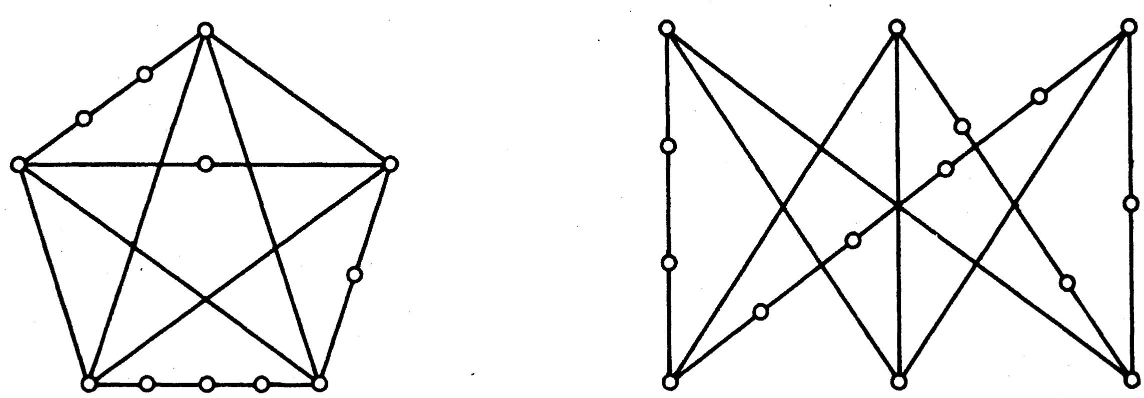
Lemma 9.10.1 If G is nonplanar, then every subdivision of G is nonplanar.

Lemma 9.10.2 If G is planar, then every subgraph of G is planar.

Since K_5 and $K_{3,3}$ are nonplanar, we see from these two lemmas that if G is planar, then G cannot contain a subdivision of K_5 or of $K_{3,3}$ (figure 9.16). Kuratowski showed that this necessary condition is also sufficient.

Before proving Kuratowski's theorem, we need to establish two more simple lemmas.

Let G be a graph with a 2-vertex cut $\{u, v\}$. Then there exist edge-disjoint subgraphs G_1 and G_2 such that $V(G_1) \cap V(G_2) = \{u, v\}$ and $G_1 \cup G_2 = G$. Consider such a separation of G into subgraphs. In both G_1 and G_2 join u



(a) (b)

Figure 9.16. (*a*) A subdivision of K_5; (*b*) a subdivision of $K_{3,3}$

FEUE WHEP

5731

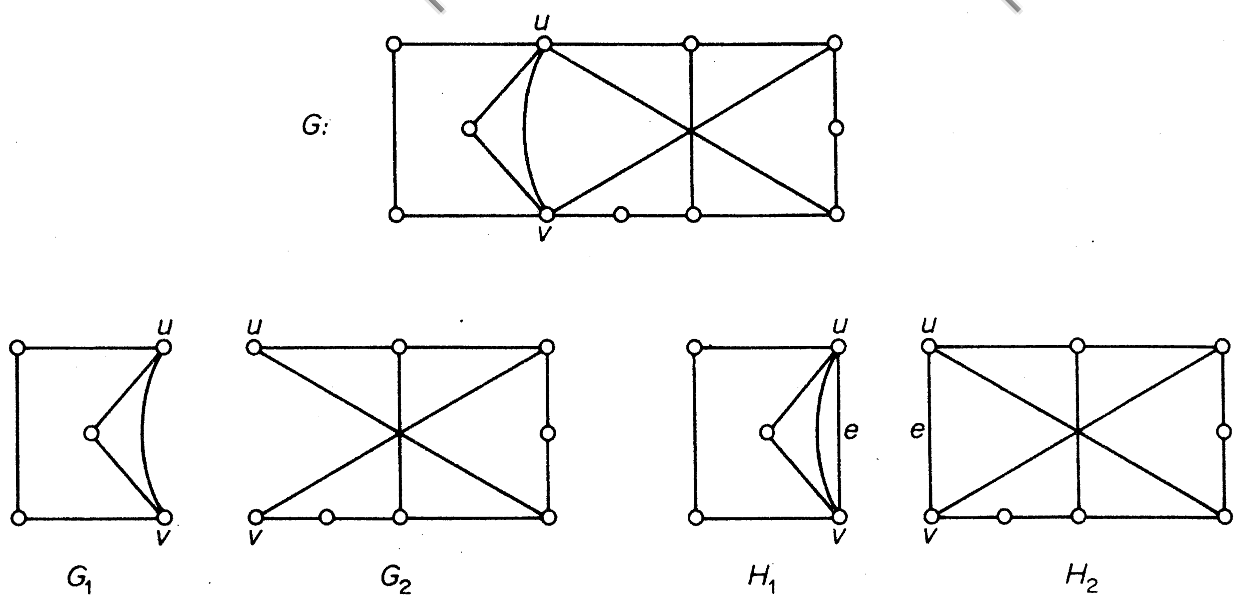


Figure 9.17

and v by a new edge e to obtain graphs H_1 and H_2, as in figure 9.17. Clearly $G=(H_1 \cup H_2)-e$. It is also easily seen that $\varepsilon(H_i)<\varepsilon(G)$ for $i=1, 2$.

Lemma 9.10.3 If G is nonplanar, then at least one of H_1 and H_2 is also nonplanar.

Proof By contradiction. Suppose that both H_1 and H_2 are planar. Let $\tilde{H}_1$ be a planar embedding of H_1, and let f be a face of $\tilde{H}_1$ incident with e. If $\tilde{H}_2$ is an embedding of H_2 in f such that $\tilde{H}_1$ and $\tilde{H}_2$ have only the vertices u and v and the edge e in common, then $(\tilde{H}_1 \cup \tilde{H}_2)-e$ is a planar embedding of G. This contradicts the hypothesis that G is nonplanar □

Lemma 9.10.4 Let G be a nonplanar connected graph that contains no subdivision of K_5 or $K_{3,3}$ and has as few edges as possible. Then G is simple and 3-connected.

Proof By contradiction. Let G satisfy the hypotheses of the lemma. Then G is clearly a minimal nonplanar graph, and therefore (exercise 9.2.1*b*) must be a simple block. If G is not 3-connected, let $\{u, v\}$ be a 2-vertex cut of G and let H_1 and H_2 be the graphs obtained from this cut as described above. By lemma 9.10.3, at least one of H_1 and H_2, say H_1, is nonplanar. Since $\varepsilon(H_1)<\varepsilon(G)$, H_1 must contain a subgraph K which is a subdivision of K_5 or $K_{3,3}$; moreover $K \not\subseteq G$, and so the edge e is in K. Let P be a (u, v)-path in H_2-e. Then G contains the subgraph $(K \cup P)-e$, which is a subdivision of K and hence a subdivision of K_5 or $K_{3,3}$. This contradiction establishes the lemma □

We shall find it convenient to adopt the following notation in the proof of Kuratowski's theorem. Suppose that C is a cycle in a plane graph. Then we

FEUE WHEP

5732

can regard the two possible orientations of C as 'clockwise' and 'anticlockwise'. For any two vertices, u and v of C, we shall denote by $C[u, v]$ the (u, v)-path which follows the clockwise orientation of C; similarly we shall use the symbols $C(u, v]$, $C[u, v)$ and $C(u, v)$ to denote the paths $C[u, v]-u$, $C[u, v]-v$ and $C[u, v]-\{u, v\}$. We are now ready to prove Kuratowski's theorem. Our proof is based on that of Dirac and Schuster (1954).

Theorem 9.10 A graph is planar if and only if it contains no subdivision of K_5 or $K_{3,3}$.

Proof We have already noted that the necessity follows from lemmas 9.10.1 and 9.10.2. We shall prove the sufficiency by contradiction.

If possible, choose a nonplanar graph G that contains no subdivision of K_5 or $K_{3,3}$ and has as few edges as possible. From lemma 9.10.4 it follows that G is simple and 3-connected. Clearly G must also be a minimal nonplanar graph.

Let uv be an edge of G, and let H be a planar embedding of the planar graph $G-uv$. Since G is 3-connected, H is 2-connected and, by corollary 3.2.1, u and v are contained together in a cycle of H. Choose a cycle C of H that contains u and v and is such that the number of edges in $\text{Int}\, C$ is as large as possible.

Since H is simple and 2-connected, each bridge of C in H must have at least two vertices of attachment. Now all outer bridges of C must be 2-bridges that overlap uv because, if some outer bridge were a k-bridge for $k \geq 3$ or a 2-bridge that avoided uv, then there would be a cycle C' containing u and v with more edges in its interior than C, contradicting the choice of C. These two cases are illustrated in figure 9.18 (with C' indicated by heavy lines).

In fact, all outer bridges of C in H must be single edges. For if a 2-bridge with vertices of attachment x and y had a third vertex, the set $\{x, y\}$ would be a 2-vertex cut of G, contradicting the fact that G is 3-connected.

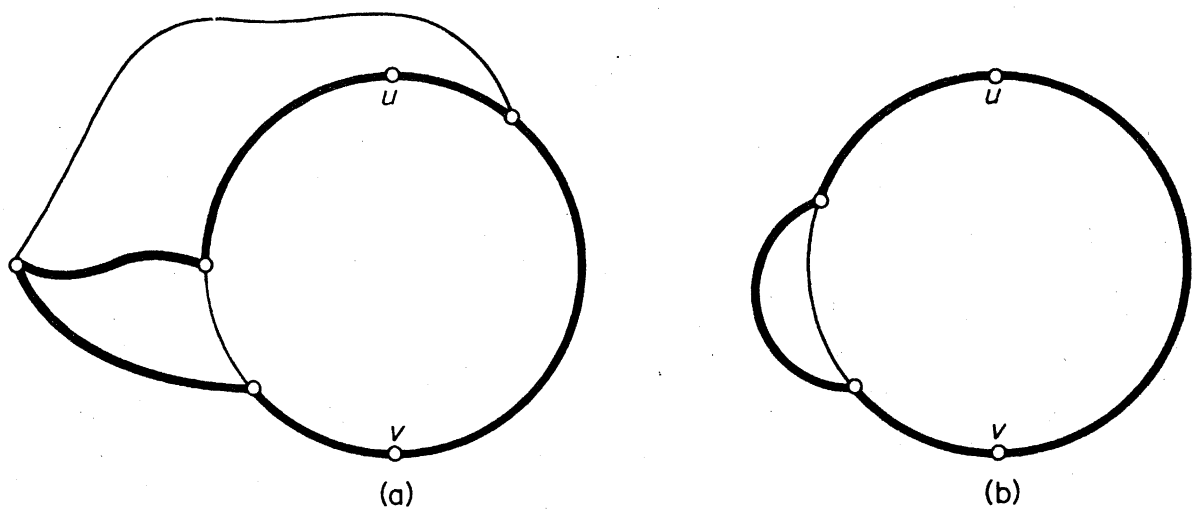


Figure 9.18

FEUE WHEP

5733

By theorem 9.8, no two inner bridges overlap. Therefore some inner bridge skew to uv must overlap some outer bridge. For otherwise, by theorem 9.9, all such bridges could be transferred (one by one), and then the edge uv could be drawn in Int C to obtain a planar embedding of G; since G is nonplanar, this is not possible. Therefore, there is an inner bridge B that is both skew to uv and skew to some outer bridge xy.

Two cases now arise, depending on whether B has a vertex of attachment different from u, v, x and y or not.

Case 1 B has a vertex of attachment different from u, v, x and y. We can choose the notation so that B has a vertex of attachment v_1 in $C(x, u)$ (see figure 9.19). We consider two subcases, depending on whether B has a vertex of attachment in $C(y, v)$ or not.

Case 1a B has a vertex of attachment v_2 in $C(y, v)$. In this case there is a (v_1, v_2)-path P in B that is internally-disjoint from C. But then $(C \cup P)+\{uv, xy\}$ is a subdivision of $K_{3,3}$ in G, a contradiction (see figure 9.19).

Case 1b B has no vertex of attachment in $C(y, v)$. Since B is skew to uv and to xy, B must have vertices of attachment v_2 in $C(u, y]$ and v_3 in $C[v, x)$. Thus B has three vertices of attachment v_1, v_2 and v_3. By theorem 9.7, there exists a vertex v_0 in $V(B)\backslash V(C)$ and three paths P_1, P_2 and P_3 in B joining v_0 to v_1, v_2 and v_3, respectively, such that, for $i \neq j$, P_i and P_j have only the vertex v_0 in common. But now $(C \cup P_1 \cup P_2 \cup P_3)+\{uv, xy\}$ contains a subdivision of $K_{3,3}$, a contradiction. This case is illustrated in figure 9.20. The subdivision of $K_{3,3}$ is indicated by heavy lines.

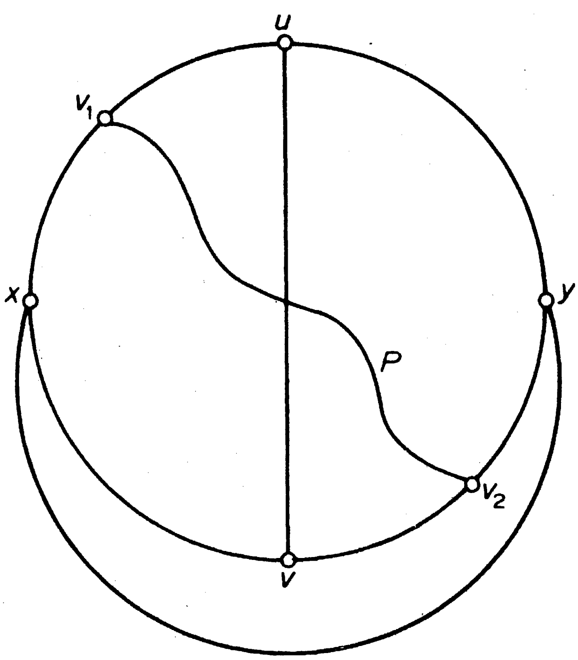


Figure 9.19

FEUE WHEP

5734

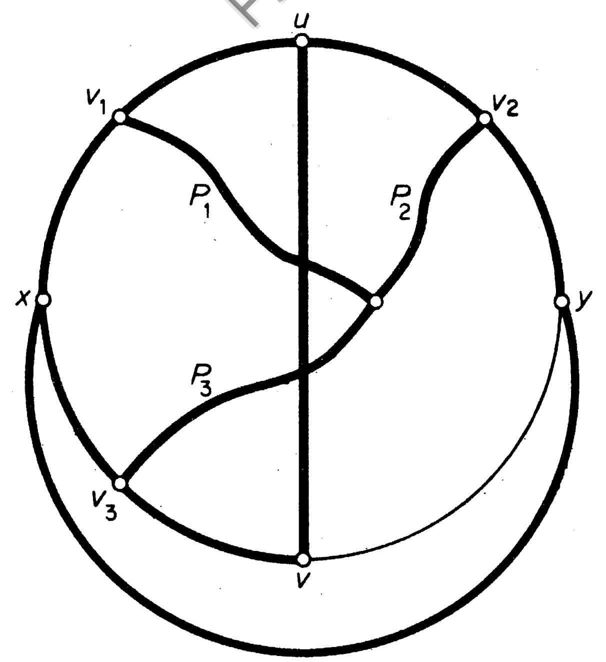


Figure 9.20

Case 2 B has no vertex of attachment other than u, v, x and y. Since B is skew to both uv and xy, it follows that u, v, x and y must all be vertices of attachment of B. Therefore (exercise 9.4.2) there exists a (u, v)-path P and an (x, y)-path Q in B such that (i) P and Q are internally-disjoint from C, and (ii) $|V(P) \cap V(Q)| \geq 1$. We consider two subcases, depending on whether P and Q have one or more vertices in common.

Case 2a $|V(P) \cap V(Q)| = 1$. In this case $(C \cup P \cup Q) + \{uv, xy\}$ is a subdivision of K_5 in G, again a contradiction (see figure 9.21).

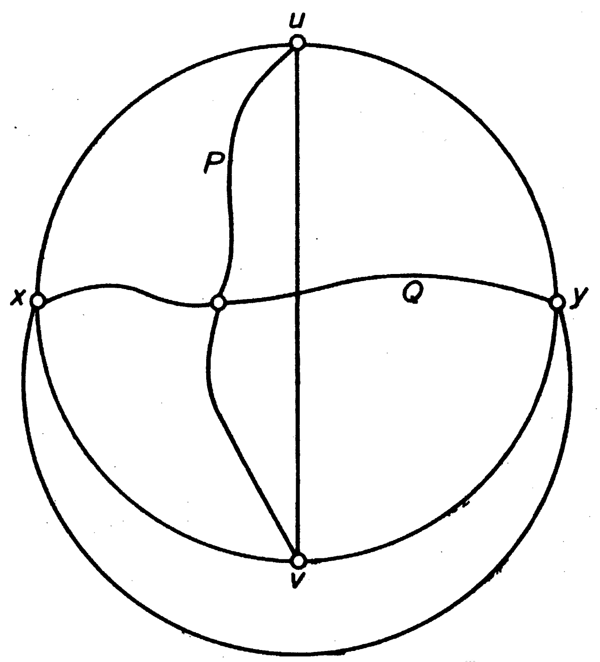


Figure 9.21

5735

Case 2b $|V(P)\cap V(Q)|\geq 2$. Let u' and v' be the first and last vertices of P on Q, and let P_1 and P_2 denote the (u, u')- and (v', v)-sections of P. Then $(C\cup P_1\cup P_2\cup Q)+\{uv, xy\}$ contains a subdivision of $K_{3,3}$ in G, once more a contradiction (see figure 9.22).

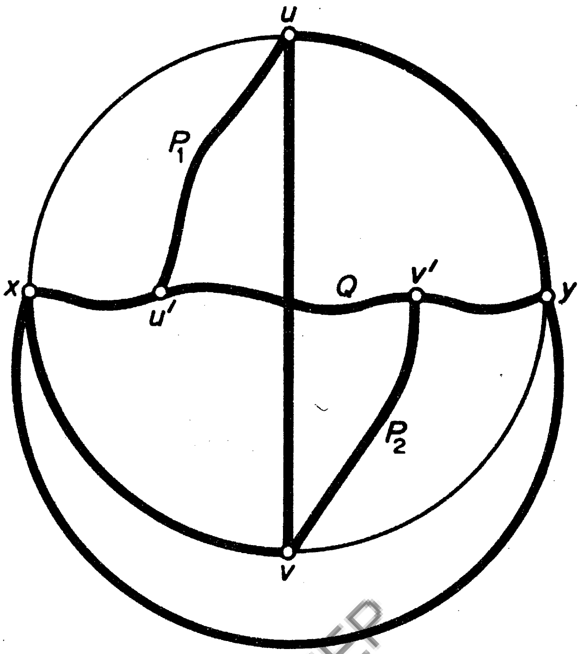


Figure 9.22

Thus all the possible cases lead to contradictions, and the proof is complete □

There are several other characterisations of planar graphs. For example, Wagner (1937) has shown that a graph is planar if and only if it contains no subgraph contractible to K_5 or $K_{3,3}$.

Exercises

9.5.1 Prove lemmas 9.10.1 and 9.10.2.

9.5.2 Show, using Kuratowski's theorem, that the Petersen graph is nonplanar.

9.6 THE FIVE-COLOUR THEOREM AND THE FOUR-COLOUR CONJECTURE

As has already been noted (exercise 9.3.2), every planar graph is 6-vertex-colourable. Heawood (1890) improved upon this result by showing that one can always properly colour the vertices of a planar graph with at most five colours. This is known as the *five-colour theorem.*

Theorem 9.11 Every planar graph is 5-vertex-colourable.

Proof By contradiction. Suppose that the theorem is false. Then there exists a 6-critical plane graph G. Since a critical graph is simple, we see from

5736

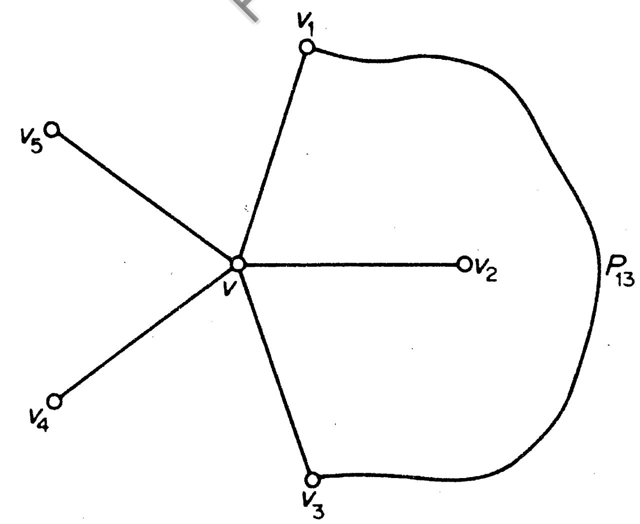


Figure 9.23

corollary 9.5.3 that $\delta \leq 5$. On the other hand we have, by theorem 8.1, that $\delta \geq 5$. Therefore $\delta = 5$. Let v be a vertex of degree five in G, and let $(V_1, V_2, V_3, V_4, V_5)$ be a proper 5-vertex colouring of $G - v$; such a colouring exists because G is 6-critical. Since G itself is not 5-vertex-colourable, v must be adjacent to a vertex of each of the five colours. Therefore we can assume that the neighbours of v in clockwise order about v are v_1, v_2, v_3, v_4 and v_5, where $v_i \in V_i$ for $1 \leq i \leq 5$.

Denote by G_{ij} the subgraph $G[V_i \cup V_j]$ induced by $V_i \cup V_j$. Now v_i and v_j must belong to the same component of G_{ij}. For, otherwise, consider the component of G_{ij} that contains v_i. By interchanging the colours i and j in this component, we obtain a new proper 5-vertex colouring of $G - v$ in which only four colours (all but i) are assigned to the neighbours of v. We have already shown that this situation cannot arise. Therefore v_i and v_j must belong to the same component of G_{ij}. Let P_{ij} be a (v_i, v_j)-path in G_{ij}, and let C denote the cycle $vv_1P_{13}v_3v$ (see figure 9.23).

Since C separates v_2 and v_4 (in figure 9.23, $v_2 \in \operatorname{int} C$ and $v_4 \in \operatorname{ext} C$), it follows from the Jordan curve theorem that the path P_{24} must meet C in some point. Because G is a plane graph, this point must be a vertex. But this is impossible, since the vertices of P_{24} have colours 2 and 4, whereas no vertex of C has either of these colours □

The question now arises as to whether the five-colour theorem is best possible. It has been conjectured that every planar graph is 4-vertex-colourable; this is known as the *four-colour conjecture*. The four-colour conjecture has remained unsettled for more than a century, despite many attempts by major mathematicians to solve it. If it were true, then it would, of course, be best possible because there do exist planar graphs which are not 3-vertex-colourable (K_4 is the simplest such graph). For a history of the four-colour conjecture, see Ore (1967)†.

† The four-colour conjecture has now been settled in the affirmative by K. Appel and W. Haken; see page 253.

5737

The problem of deciding whether the four-colour conjecture is true or false is called the *four-colour problem*.† There are several problems in graph theory that are equivalent to the four-colour problem; one of these is the case $n = 5$ of Hadwiger's conjecture (see section 8.3). We now establish the equivalence of certain problems concerning edge and face colourings with the four-colour problem. A *k-face colouring* of a plane graph G is an assignment of k colours $1, 2, \ldots, k$ to the faces of G; the colouring is *proper* if no two faces that are separated by an edge have the same colour. G is *k-face-colourable* if it has a proper k-face colouring, and the minimum k for which G is k-face-colourable is the *face chromatic number* of G, denoted by $\chi^*(G)$. It follows immediately from these definitions that, for any plane graph G with dual G^*,

$$\chi^*(G) = \chi(G^*) \tag{9.2}$$

Theorem 9.12 The following three statements are equivalent:

(i) every planar graph is 4-vertex-colourable;
(ii) every plane graph is 4-face-colourable;
(iii) every simple 2-edge-connected 3-regular planar graph is 3-edge-colourable.

Proof We shall show that (i)⇒(ii)⇒(iii)⇒(i).

(*a*) (i)⇒(ii). This is a direct consequence of (9.2) and the fact that the dual of a plane graph is planar.

(*b*) (ii)⇒(iii). Suppose that (ii) holds, let G be a simple 2-edge-connected 3-regular planar graph, and let $\tilde{G}$ be a planar embedding of G. By (ii), $\tilde{G}$ has a proper 4-face-colouring. It is, of course, immaterial which symbols are used as the 'colours', and in this case we shall denote the four colours by the vectors $c_0 = (0, 0)$, $c_1 = (1, 0)$, $c_2 = (0, 1)$ and $c_3 = (1, 1)$, over the field of integers modulo 2. We now obtain a 3-edge-colouring of $\tilde{G}$ by assigning to each edge the sum of the colours of the faces it separates (see figure 9.24). If c_i, c_j and c_k are the three colours assigned to the three faces incident with a vertex v, then $c_i + c_j$, $c_j + c_k$ and $c_k + c_i$ are the colours assigned to the three edges incident with v. Since $\tilde{G}$ is 2-edge-connected, each edge separates two distinct faces, and it follows that no edge is assigned the colour c_0 under this scheme. It is also clear that the three edges incident with a given vertex are assigned different colours. Thus we have a proper 3-edge-colouring of $\tilde{G}$, and hence of G.

† The four-colour problem is often posed in the following terms: can the countries of any map be coloured in four colours so that no two countries which have a common boundary are assigned the same colour? The equivalence of this problem with the four-colour problem follows from theorem 9.12 on observing that a map can be regarded as a plane graph with its countries as the faces.

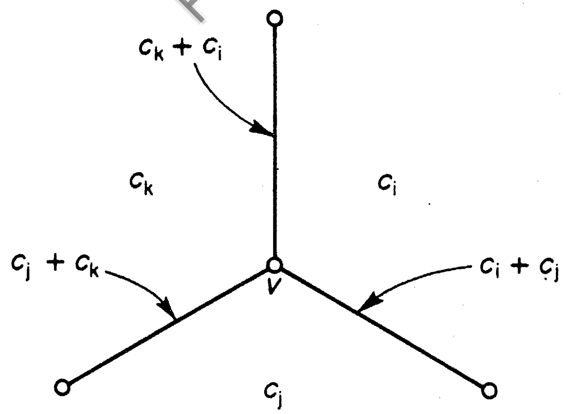


Figure 9.24

(*c*) (iii)⇒(i). Suppose that (iii) holds, but that (i) does not. Then there is a 5-critical planar graph G. Let $\tilde{G}$ be a planar embedding of G. Then (exercise 9.2.6) $\tilde{G}$ is a spanning subgraph of a simple plane triangulation H. The dual H^* of H is a simple 2-edge-connected 3-regular planar graph (exercise 9.2.7). By (iii), H^* has a proper 3-edge colouring (E_1, E_2, E_3). For $i \neq j$, let H^*_{ij} denote the subgraph of H^* induced by $E_i \cup E_j$. Since each vertex of H^* is incident with one edge of E_i and one edge of E_j, H^*_{ij} is a union of disjoint cycles and is therefore (exercise 9.6.1) 2-face-colourable. Now each face of H^* is the intersection of a face of H^*_{12} and a face of H^*_{23}. Given proper 2-face colourings of H^*_{12} and H^*_{23} we can obtain a 4-face colouring of H^* by assigning to each face f the pair of colours assigned to the faces whose intersection is f. Since $H^* = H^*_{12} \cup H^*_{23}$ it is easily verified that this 4-face colouring of H^* is proper. Since H is a supergraph of G we have

$$5 = \chi(G) \leq \chi(H) = \chi^*(H^*) \leq 4$$

This contradiction shows that (i) does, in fact, hold □

That statement (iii) of theorem 9.12 is equivalent to the four-colour problem was first observed by Tait (1880). A proper 3-edge colouring of a 3-regular graph is often called a *Tait colouring*. In the next section we shall discuss Tait's ill-fated approach to the four-colour conjecture. Grötzsch (1958) has verified the four-colour conjecture for planar graphs without triangles. In fact, he has shown that every such graph is 3-vertex-colourable.

Exercises

9.6.1 Show that a plane graph G is 2-face-colourable if and only if G is eulerian.

9.6.2 Show that a plane triangulation G is 3-vertex colourable if and only if G is eulerian.

9.6.3 Show that every hamiltonian plane graph is 4-face-colourable.

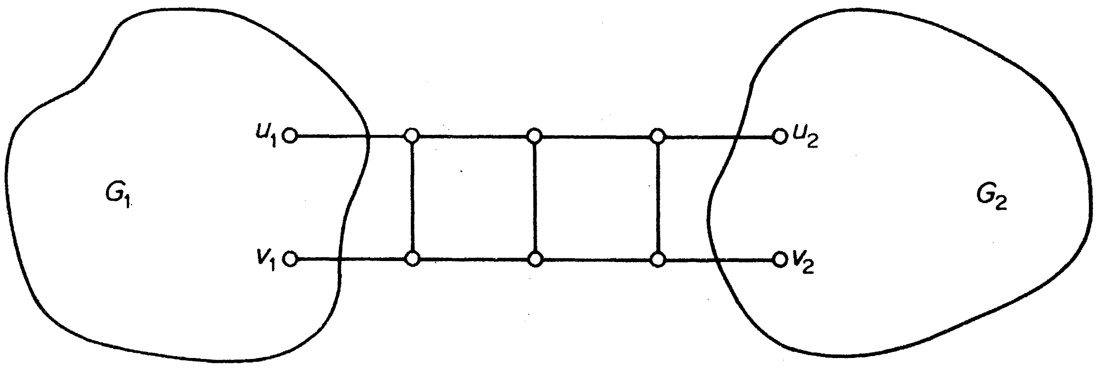
9.6.4 Show that every hamiltonian 3-regular graph has a Tait colouring.

5739

9.6.5 Prove theorem 9.12 by showing that (iii)$\Rightarrow$(ii)$\Rightarrow$(i)$\Rightarrow$(iii).

9.6.6 Let G be a 3-regular graph with $\kappa' = 2$.

(*a*) Show that there exist subgraphs G_1 and G_2 of G and nonadjacent pairs of vertices $u_1, v_1 \in V(G_1)$ and $u_2, v_2 \in V(G_2)$ such that G consists of the graphs G_1 and G_2 joined by a 'ladder' at the vertices u_1, v_1, u_2 and v_2.



(*b*) Show that if $G_1 + u_1v_1$ and $G_2 + u_2v_2$ both have Tait colourings, then so does G.

(*c*) Deduce, using theorem 9.12, that the four-colour conjecture is equivalent to *Tait's conjecture:* every simple 3-regular 3-connected planar graph has a Tait colouring.

9.6.7 Give an example of

(*a*) a 3-regular planar graph with no Tait colouring;

(*b*) a 3-regular 2-connected graph with no Tait colouring.

9.7 NONHAMILTONIAN PLANAR GRAPHS

In his attempt to prove the four-colour conjecture, Tait (1880) observed that it would be enough to show that every 3-regular 3-connected planar graph has a Tait colouring (exercise 9.6.6). By mistakenly assuming that every such graph is hamiltonian, he gave a 'proof' of the four-colour conjecture (see exercise 9.6.4). Over half a century later, Tutte (1946) showed Tait's proof to be invalid by constructing a nonhamiltonian 3-regular 3-connected planar graph; it is depicted in figure 9.25.

Tutte proved that his graph is nonhamiltonian by using ingenious *ad hoc* arguments (exercise 9.7.1), and for many years the Tutte graph was the only known example of a nonhamiltonian 3-regular 3-connected planar graph. However, Grinberg (1968) then discovered a necessary condition for a plane graph to be hamiltonian. His discovery has led to the construction of many nonhamiltonian planar graphs.

FEUE WHEP

5740

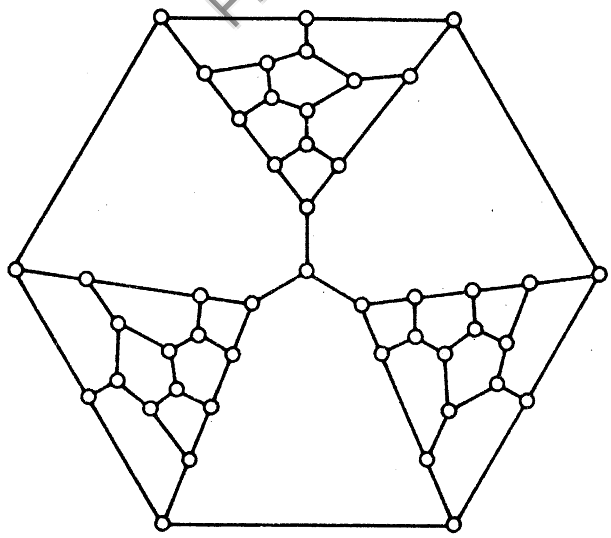


Figure 9.25. The Tutte graph

Theorem 9.13 Let G be a loopless plane graph with a Hamilton cycle C. Then

$$\sum_{i=1}^{\nu} (i-2)(\phi_i' - \phi_i'') = 0 \tag{9.3}$$

where ϕ_i' and ϕ_i'' are the numbers of faces of degree i contained in Int C and Ext C, respectively.

Proof Denote by E' the subset of $E(G)\backslash E(C)$ contained in Int C, and let $\varepsilon' = |E'|$. Then Int C contains exactly $\varepsilon' + 1$ faces (see figure 9.26), and so

$$\sum_{i=1}^{\nu} \phi_i' = \varepsilon' + 1 \tag{9.4}$$

Now each edge in E' is on the boundary of two faces in Int C, and each edge

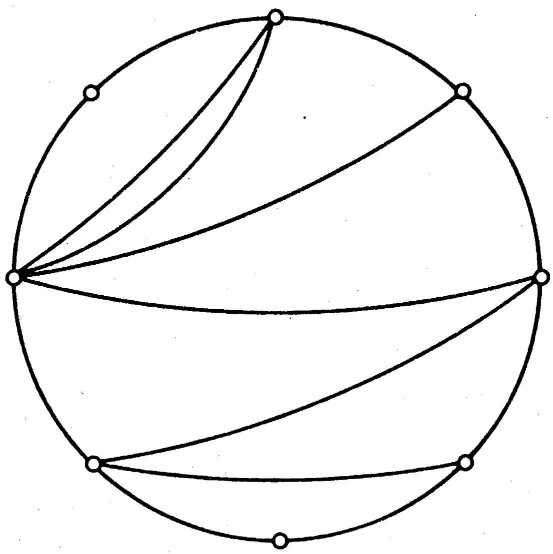


Figure 9.26

FEUE WHEP

5741

of C is on the boundary of exactly one face in Int C. Therefore

$$\sum_{i=1}^{\nu} i\phi_i' = 2\varepsilon' + \nu \tag{9.5}$$

Using (9.4), we can eliminate ε' from (9.5) to obtain

$$\sum_{i=1}^{\nu} (i-2)\phi_i' = \nu - 2 \tag{9.6}$$

Similarly

$$\sum_{i=1}^{\nu} (i-2)\phi_i'' = \nu - 2 \tag{9.7}$$

Equations (9.6) and (9.7) now yield (9.3) □

With the aid of theorem 9.13, it is a simple matter to show, for example, that the Grinberg graph (figure 9.27) is nonhamiltonian.

Suppose that this graph is hamiltonian. Then, noting that it only has faces of degrees five, eight and nine, condition (9.3) yields

$$3(\phi_5' - \phi_5'') + 6(\phi_8' - \phi_8'') + 7(\phi_9' - \phi_9'') = 0$$

We deduce that

$$7(\phi_9' - \phi_9'') \equiv 0 \quad (\text{modulo } 3)$$

But this is clearly impossible, since the value of the left-hand side is 7 or −7, depending on whether the face of degree nine is in Int C or Ext C. Therefore the graph cannot be hamiltonian.

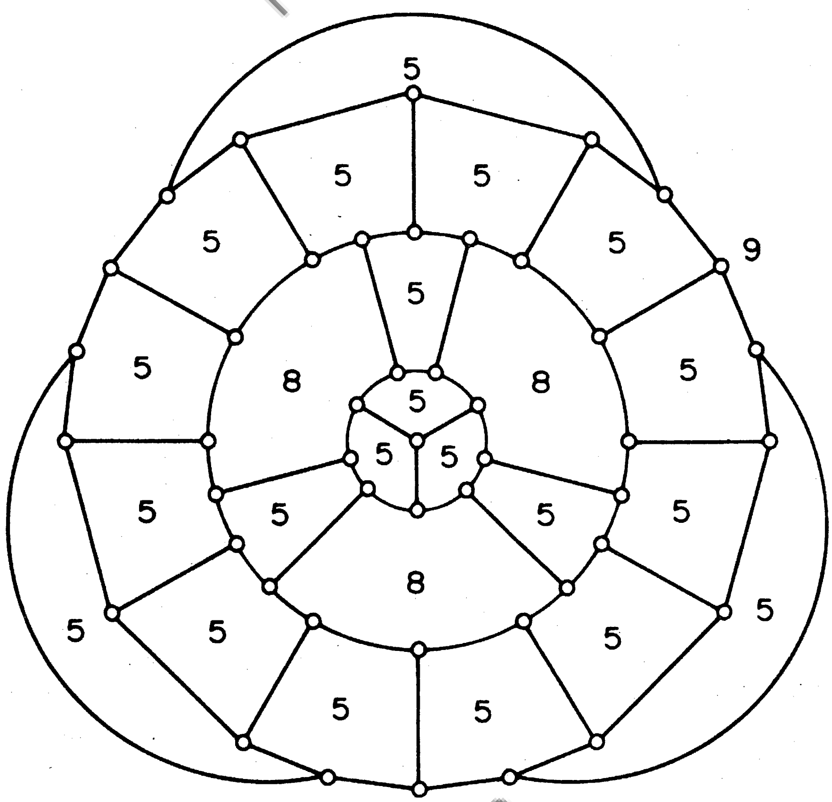


Figure 9.27. The Grinberg graph

FEUE WHEP

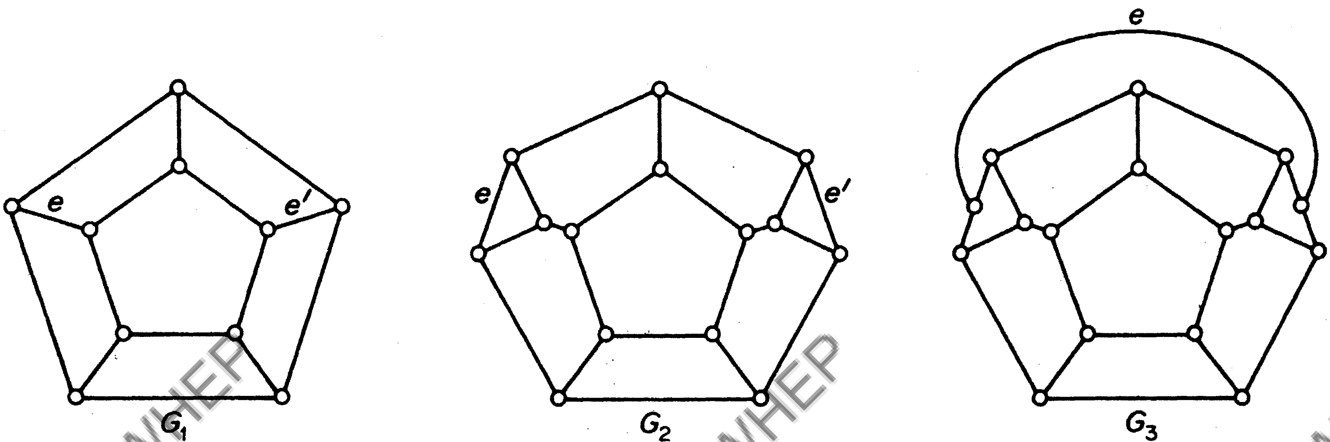
Although there exist nonhamiltonian 3-connected planar graphs, Tutte (1956) has shown that every 4-connected planar graph is hamiltonian.

Exercises

9.7.1 (*a*) Show that no Hamilton cycle in the graph G_1 below can contain both the edges e and e'.

(*b*) Using (*a*), show that no Hamilton cycle in the graph G_2 can contain both the edges e and e'.

(*c*) Using (*b*), show that every Hamilton cycle in the graph G_3 must contain the edge e.



(*d*) Deduce that the Tutte graph (figure 9.25) is nonhamiltonian.

9.7.2 Show, by applying theorem 9.13, that the Herschel graph (figure 4.2*b*) is nonhamiltonian. (It is, in fact, the smallest nonhamiltonian 3-connected planar graph.)

9.7.3 Give an example of a simple nonhamiltonian 3-regular planar graph with connectivity two.

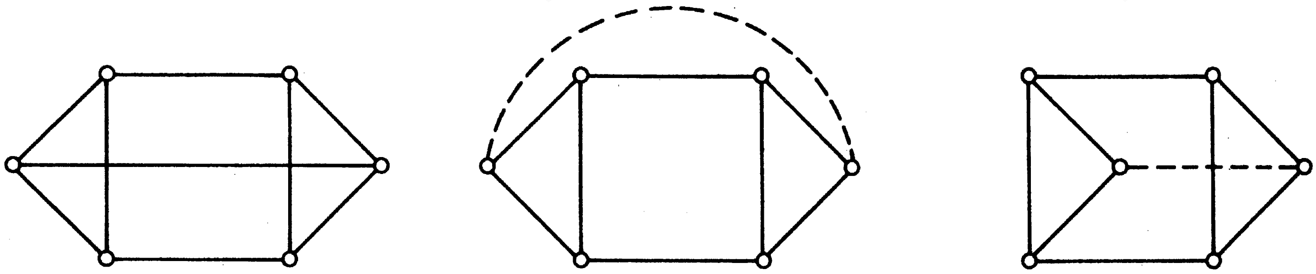
APPLICATIONS

9.8 A PLANARITY ALGORITHM

There are many practical situations in which it is important to decide whether a given graph is planar, and, if so, to then find a planar embedding of the graph. For example, in the layout of printed circuits one is interested in knowing if a particular electrical network is planar. In this section, we shall present an algorithm for solving this problem, due to Demoucron, Malgrange and Pertuiset (1964).

Let H be a planar subgraph of a graph G and let $\tilde{H}$ be an embedding of H in the plane. We say that $\tilde{H}$ is *G-admissible* if G is planar and there is a planar embedding $\tilde{G}$ of G such that $\tilde{H} \subseteq \tilde{G}$. In figure 9.28, for example, two embeddings of a planar subgraph of G are shown; one is G-admissible and the other is not.

5743



(a) (b) (c)

Figure 9.28. (*a*) G; (*b*) G-admissible; (*c*) G-inadmissible

If B is any bridge of H (in G), then B is said to be *drawable* in a face f of $\tilde{H}$ if the vertices of attachment of B to H are contained in the boundary of f. We write $F(B, \tilde{H})$ for the set of faces of $\tilde{H}$ in which B is drawable. The following theorem provides a necessary condition for G to be planar.

Theorem 9.14 If $\tilde{H}$ is G-admissible then, for every bridge B of H, $F(B, \tilde{H}) \neq \emptyset$.

Proof If $\tilde{H}$ is G-admissible then, by definition, there exists a planar embedding $\tilde{G}$ of G such that $\tilde{H} \subseteq \tilde{G}$. Clearly, the subgraph of $\tilde{G}$ which corresponds to a bridge B of H must be confined to one face of $\tilde{H}$. Hence $F(B, \tilde{H}) \neq \emptyset$ □

Since a graph is planar if and only if each block of its underlying simple graph is planar, it suffices to consider simple blocks. Given such a graph G, the algorithm determines an increasing sequence $G_1, G_2, \ldots$ of planar subgraphs of G, and corresponding planar embeddings $\tilde{G}_1, \tilde{G}_2, \ldots$. When G is planar, each $\tilde{G}_i$ is G-admissible and the sequence $\tilde{G}_1, \tilde{G}_2, \ldots$ terminates in a planar embedding of G. At each stage, the necessary condition in theorem 9.14 is used to test G for nonplanarity.

Planarity Algorithm

1. Let G_1 be a cycle in G. Find a planar embedding $\tilde{G}_1$ of G_1. Set $i = 1$.
2. If $E(G)\backslash E(G_i) = \emptyset$, stop. Otherwise, determine all bridges of G_i in G; for each such bridge B find the set $F(B, \tilde{G}_i)$.
3. If there exists a bridge B such that $F(B, \tilde{G}_i) = \emptyset$, stop; by theorem 9.14, G is nonplanar. If there exists a bridge B such that $|F(B, \tilde{G}_i)| = 1$, let $\{f\} = F(B, \tilde{G}_i)$. Otherwise, let B be any bridge and f any face such that $f \in F(B, \tilde{G}_i)$.
4. Choose a path $P_i \subseteq B$ connecting two vertices of attachment of B to G_i. Set $G_{i+1} = G_i \cup P_i$ and obtain a planar embedding $\tilde{G}_{i+1}$ of G_{i+1} by drawing P_i in the face f of $\tilde{G}_i$. Replace i by $i+1$ and go to step 2.

To illustrate this algorithm, we shall consider the graph G of figure 9.29. We start with the cycle $\tilde{G}_1 = 2345672$ and a list of its bridges (denoted, for

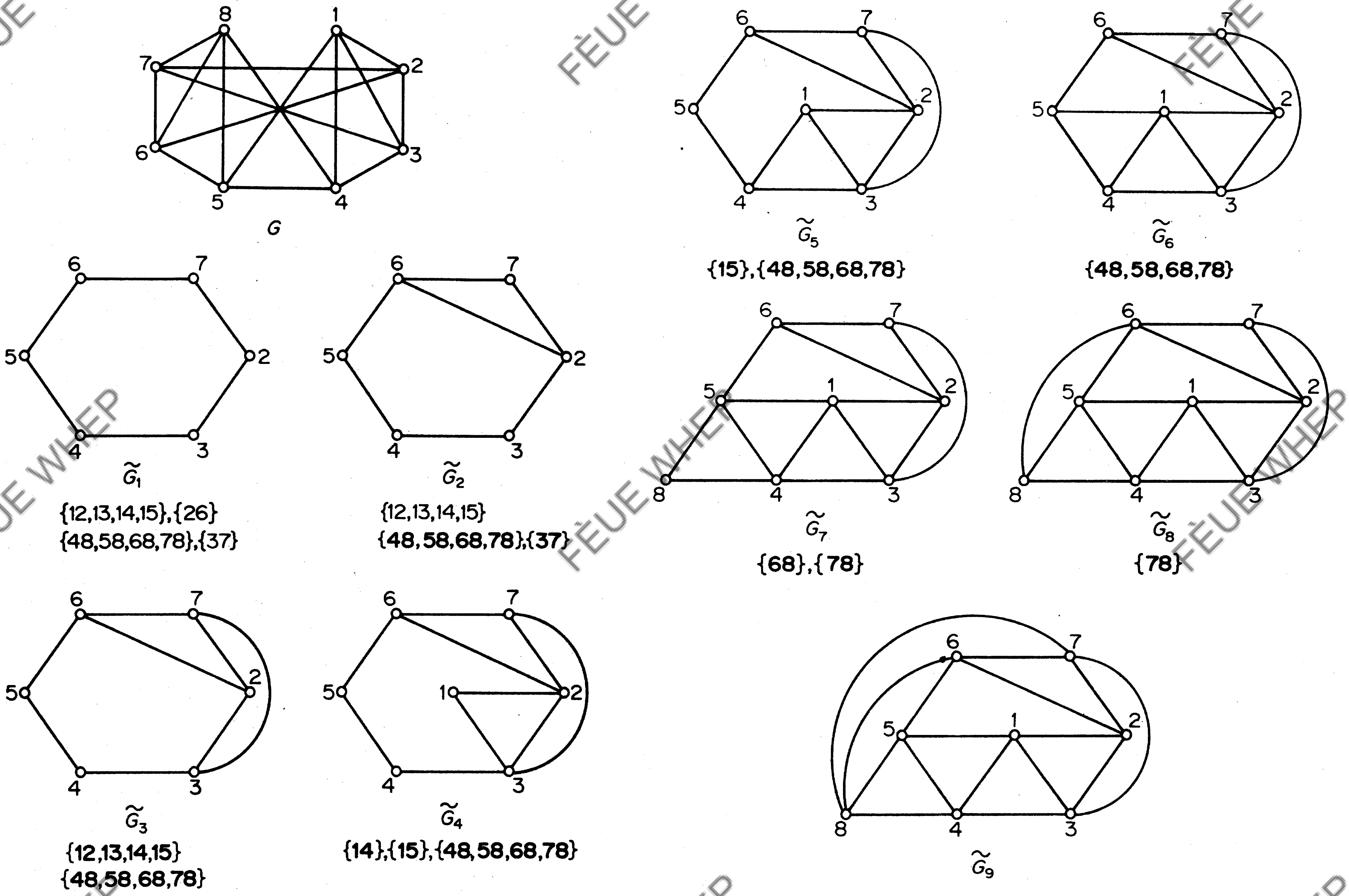

G
$\widetilde{G}_1$
{12,13,14,15},{26}
{48,58,68,78},{37}
$\widetilde{G}_2$
{12,13,14,15}
{48,58,68,78},{37}
$\widetilde{G}_3$
{12,13,14,15}
{48,58,68,78}
$\widetilde{G}_4$
{14},{15},{48,58,68,78}
$\widetilde{G}_5$
{15},{48,58,68,78}
$\widetilde{G}_6$
{48,58,68,78}
$\widetilde{G}_7$
{68},{78}
$\widetilde{G}_8$
{78}
$\widetilde{G}_9$

Figure 9.29

5745

brevity, by their edge sets); at each stage, the bridges B for which $|F(B, \tilde{G}_i)| = 1$ are indicated in bold face. In this example, the algorithm terminates with a planar embedding $\tilde{G}_9$ of G. Thus G is planar.

Now let us apply the algorithm to the graph H obtained from G by deleting edge 45 and adding edge 36 (figure 9.30). Starting with the cycle 23672, we proceed as shown in figure 9.30. It can be seen that, having constructed $\tilde{H}_3$, we find a bridge $B = \{12, 13, 14, 15, 34, 48, 56, 58, 68, 78\}$

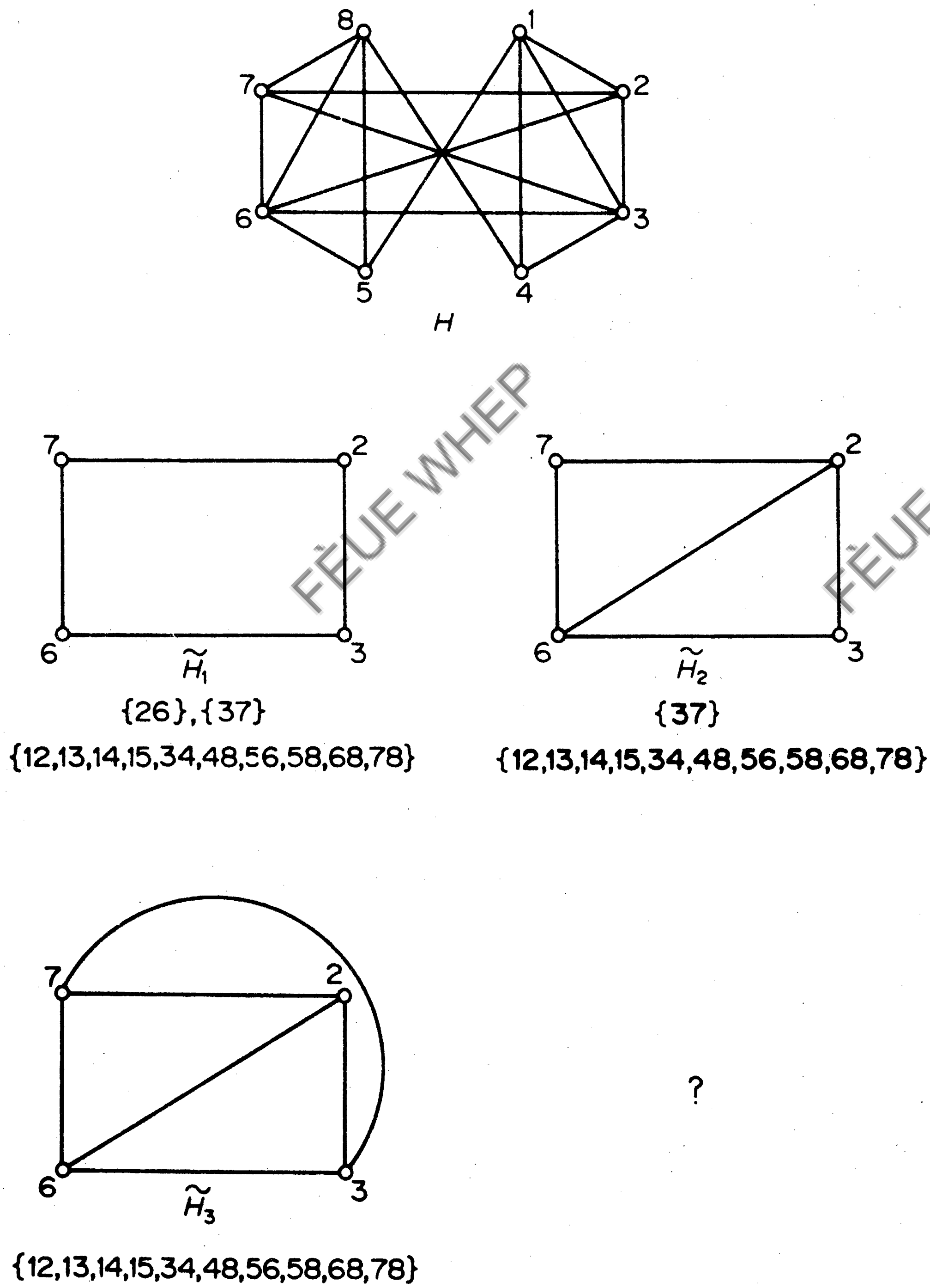


Figure 9.30

FEUE WHEP

5746

such that $F(B, \tilde{H}_3) = \emptyset$. At this point the algorithm stops (step 3), and we conclude that H is nonplanar.

In order to establish the validity of the algorithm, one needs to show that if G is planar, then each term of the sequence $\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_{\varepsilon-\nu+1}$ is G-admissible. Demoucron, Malgrange and Pertuiset prove this by induction. We shall give a general outline of their proof.

Suppose that G is planar. Clearly $\tilde{G}_1$ is G-admissible. Assume that $\tilde{G}_i$ is G-admissible for $1 \le i \le k < \varepsilon - \nu + 1$. By definition, there is a planar embedding $\tilde{G}$ of G such that $\tilde{G}_k \subset \tilde{G}$. We wish to show that $\tilde{G}_{k+1}$ is G-admissible. Let B and f be as defined in step 3 of the algorithm. If, in $\tilde{G}$, B is drawn in f, $\tilde{G}_{k+1}$ is clearly G-admissible. So assume that no bridge of G_k is drawable in only one face of $\tilde{G}_k$, and that, in $\tilde{G}$, B is drawn in some other face f'. Since no bridge is drawable in just one face, no bridge whose vertices of attachment are restricted to the common boundary of f and f' can be skew to a bridge not having this property. Hence we can interchange bridges across the common boundary of f and f' and thereby obtain a planar embedding of G in which B is drawn in f (see figure 9.31). Thus, again, $\tilde{G}_{k+1}$ is G-admissible.

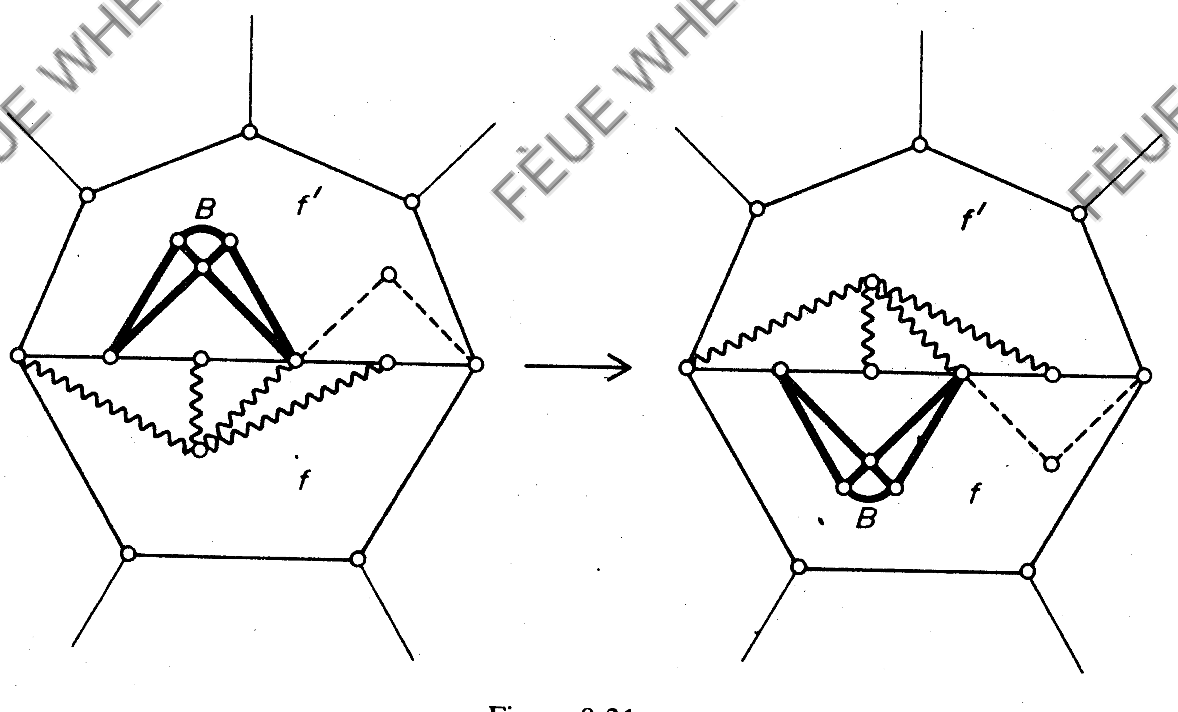


Figure 9.31

The algorithm that we have described is good. From the flow diagram (figure 9.32), one sees that the main operations involved are

(i) finding a cycle G_1 in the block G;
(ii) determining the bridges of G_i in G and their vertices of attachment to G_i;

5747

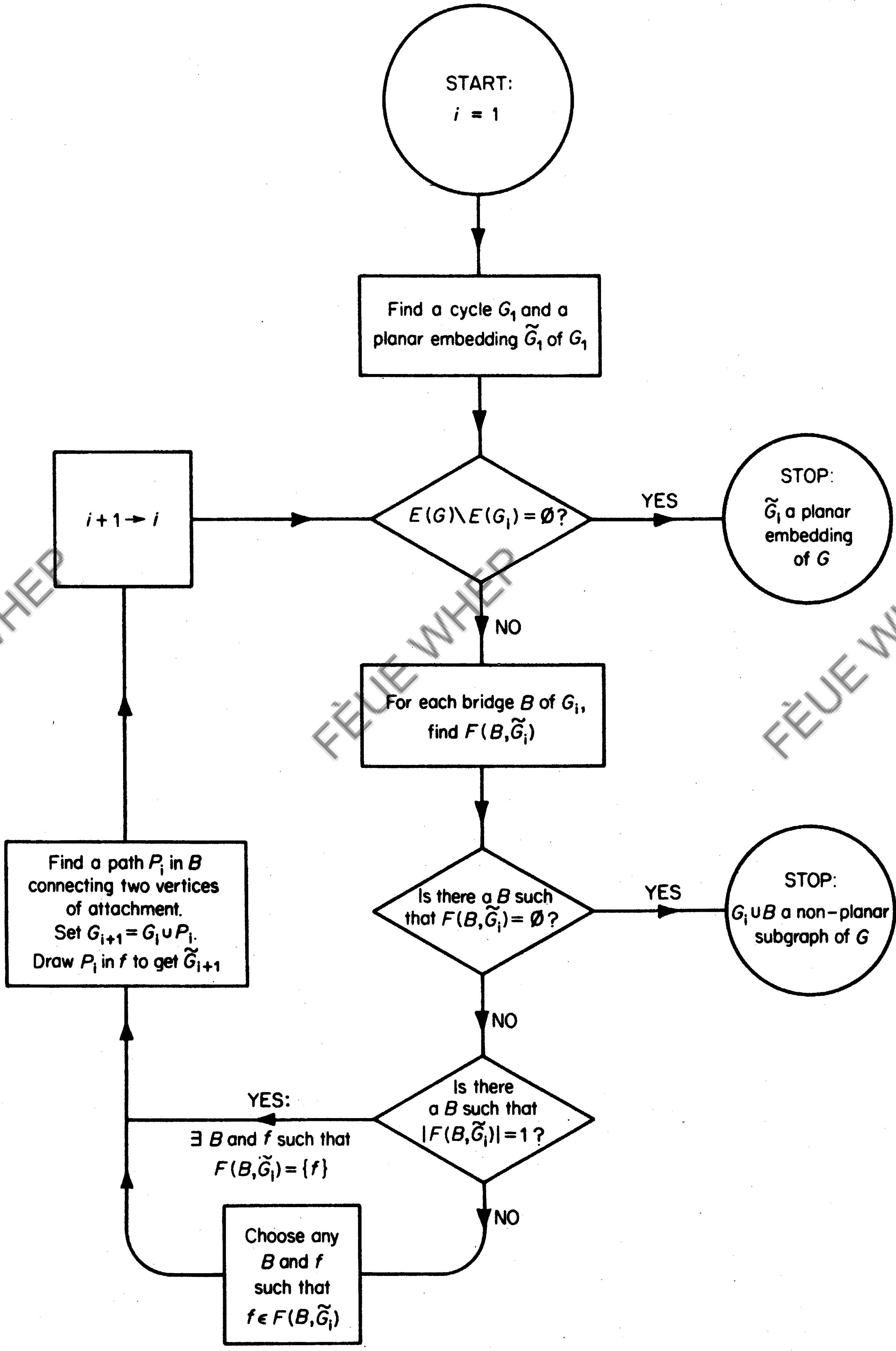


Figure 9.32. Planarity algorithm

5748

(iii) determining $b(f)$ for each face f of $\tilde{G}_i$;
(iv) determining $F(B, \tilde{G}_i)$ for each bridge B of G_i;
(v) finding a path P_i in some bridge B of G_i between two vertices of $V(B, G_i)$.

There exists a good algorithm for each of these operations; we leave the details as an exercise.

More sophisticated algorithms for testing planarity than the above have since been obtained. See, for example, Hopcroft and Tarjan (1974).

Exercise

9.8.1 Show that the Petersen graph is nonplanar by applying the above algorithm.

REFERENCES

Demoucron, G., Malgrange, Y. and Pertuiset, R. (1964). Graphes planaires: reconnaissance et construction de représentations planaires topologiques. *Rev. Française Recherche Opérationnelle*, **8**, 33–47

Dirac, G. A. and Schuster, S. (1954). A theorem of Kuratowski. *Nederl. Akad. Wetensch. Proc. Ser. A.*, **57**, 343–48

Fáry, I. (1948). On straight line representation of planar graphs. *Acta Sci. Math. Szeged*, **11**, 229–33

Fréchet, M. and Ky Fan (1967). *Initiation to Combinatorial Topology*, Prindle, Weber and Schmidt, Boston

Grinberg, È. Ja (1968). Plane homogeneous graphs of degree three without Hamiltonian circuits (Russian). *Latvian Math. Yearbook*, **4**, 51–58

Grötzsch, H. (1958). Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg. Math.-Nat. Reihe*, **9**, 109–19

Heawood, P. J. (1890). Map colour theorems. *Quart. J. Math.*, **24**, 332–38

Hopcroft, J. E. and Tarjan, R. E. (1974). Efficient planarity testing. *J. Assoc. Comput. Mach.*, **21**, 549–568

Kuratowski, C. (1930). Sur le problème des courbes gauches en topologie. *Fund. Math.*, **15**, 271–83

Ore, O. (1967). *The Four-Color Problem*, Academic Press, New York

Tait, P. G. (1880). Remarks on colouring of maps. *Proc. Royal Soc. Edinburgh Ser. A.*, **10**, 729

5749

Tutte, W. T. (1946). On Hamiltonian circuits, *J. London Math. Soc.*, **21,** 98–101

Tutte, W. T. (1956). A theorem on planar graphs. *Trans. Amer. Math. Soc.*, **82,** 99–116

Wagner, K. (1937). Über eine Eigenschaft der ebenen Komplexe. *Math. Ann.*, **114,** 570–90

FEUE WHEP

5750

10 Directed Graphs

10.1 DIRECTED GRAPHS

Although many problems lend themselves naturally to a graph-theoretic formulation, the concept of a graph is sometimes not quite adequate. When dealing with problems of traffic flow, for example, it is necessary to know which roads in the network are one-way, and in which direction traffic is permitted. Clearly, a graph of the network is not of much use in such a situation. What we need is a graph in which each link has an assigned orientation—a directed graph. Formally, a *directed graph* D is an ordered triple $(V(D), A(D), \psi_D)$ consisting of a nonempty set $V(D)$ of *vertices*, a set $A(D)$, disjoint from $V(D)$, of *arcs*, and an *incidence function* ψ_D that associates with each arc of D an ordered pair of (not necessarily distinct) vertices of D. If a is an arc and u and v are vertices such that $\psi_D(a) = (u, v)$, then a is said to *join* u *to* v; u is the *tail* of a, and v is its *head*. For convenience, we shall abbreviate 'directed graph' to *digraph*. A digraph D' is a *subdigraph* of D if $V(D') \subseteq V(D)$, $A(D') \subseteq A(D)$ and $\psi_{D'}$ is the restriction of ψ_D to $A(D')$. The terminology and notation for subdigraphs is similar to that used for subgraphs.

With each digraph D we can associate a graph G on the same vertex set; corresponding to each arc of D there is an edge of G with the same ends. This graph is the *underlying graph* of D. Conversely, given any graph G, we can obtain a digraph from G by specifying, for each link, an order on its ends. Such a digraph is called an *orientation* of G.

Just as with graphs, digraphs have a simple pictorial representation. A digraph is represented by a diagram of its underlying graph together with arrows on its edges, each arrow pointing towards the head of the corresponding arc. A digraph and its underlying graph are shown in figure 10.1.

Every concept that is valid for graphs automatically applies to digraphs too. Thus the digraph of figure 10.1a is connected and has no cycle of length three because its underlying graph (figure 10.1b) has these properties. However, there are many concepts that involve the notion of orientation, and these apply only to digraphs.

A *directed walk* in D is a finite non-null sequence $W = (v_0, a_1, v_1, \ldots, a_k, v_k)$, whose terms are alternately vertices and arcs, such that, for $i = 1, 2, \ldots, k$, the arc a_i has head v_i and tail v_{i-1}. As with walks in graphs, a directed walk $(v_0, a_1, v_1, \ldots, a_k, v_k)$ is often represented simply by

5751

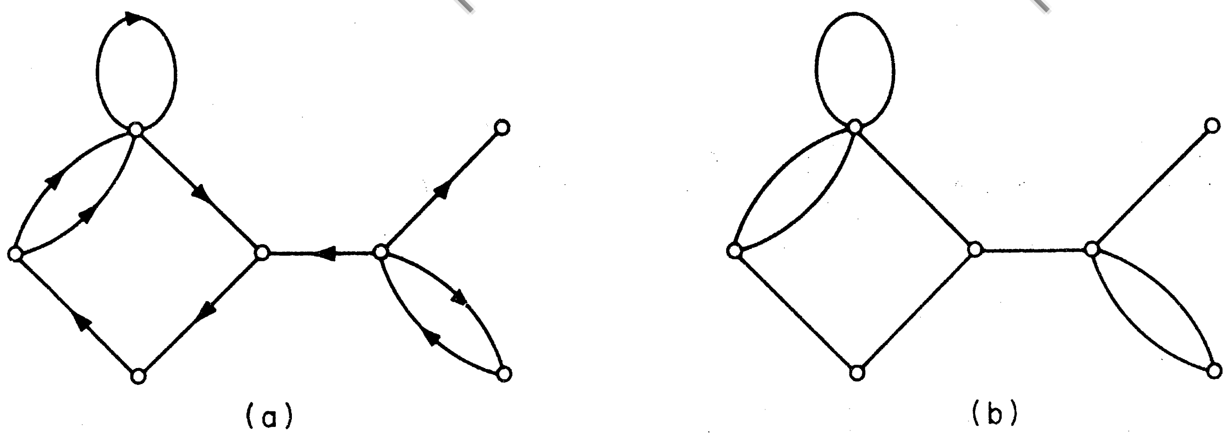


Figure 10.1. (*a*) A digraph D; (*b*) the underlying graph of D

its vertex sequence $(v_0, v_1, \ldots, v_k)$. A *directed trail* is a directed walk that is a trail; *directed paths*, *directed cycles* and *directed tours* are similarly defined.

If there is a directed (u, v)-path in D, vertex v is said to be *reachable* from vertex u in D. Two vertices are *diconnected* in D if each is reachable from the other. As in the case of connection in graphs, diconnection is an equivalence relation on the vertex set of D. The subdigraphs $D[V_1], D[V_2], \ldots, D[V_m]$ induced by the resulting partition $(V_1, V_2, \ldots, V_m)$ of $V(D)$ are called the *dicomponents* of D. A digraph D is *diconnected* if it has exactly one dicomponent. The digraph of figure 10.2*a* is not diconnected; it has the three dicomponents shown in figure 10.2*b*.

The *indegree* $d_D^-(v)$ of a vertex v in D is the number of arcs with head v; the *outdegree* $d_D^+(v)$ of v is the number of arcs with tail v. We denote the minimum and maximum indegrees and outdegrees in D by $\delta^-(D)$, $\Delta^-(D)$, $\delta^+(D)$ and $\Delta^+(D)$, respectively. A digraph is *strict* if it has no loops and no two arcs with the same ends have the same orientation.

Throughout this chapter, D will denote a digraph and G its underlying graph. This is a useful convention; it allows us, for example, to denote the vertex set of D by V (since $V = V(G)$), and the numbers of vertices and arcs in D by ν and ε, respectively. Also, as with graphs, we shall drop the letter D from our notation whenever possible; thus we write A for $A(D)$, $d^+(v)$ for $d_D^+(v)$, δ^- for $\delta^-(D)$, and so on.

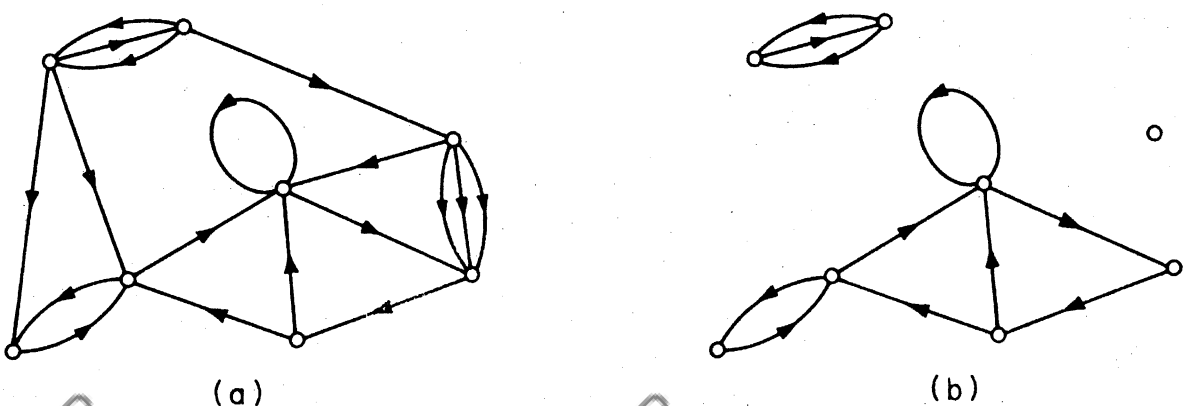


Figure 10.2. (*a*) A digraph D; (*b*) the three dicomponents of D

5752

Exercises

10.1.1 How many orientations does a simple graph G have?

10.1.2 Show that $\sum_{v \in V} d^-(v) = \varepsilon = \sum_{v \in V} d^+(v)$

10.1.3 Let D be a digraph with no directed cycle.

(*a*) Show that $\delta^- = 0$.

(*b*) Deduce that there is an ordering $v_1, v_2, \ldots, v_\nu$ of V such that, for $1 \leq i \leq \nu$, every arc of D with head v_i has its tail in $\{v_1, v_2, \ldots, v_{i-1}\}$.

10.1.4 Show that D is diconnected if and only if D is connected and each block of D is diconnected.

10.1.5 The *converse* $\overleftarrow{D}$ of D is the digraph obtained from D by reversing the orientation of each arc.

(*a*) Show that

(i) $\overleftarrow{\overleftarrow{D}} = D$;

(ii) $d^+_{\overleftarrow{D}}(v) = d^-_D(v)$;

(iii) v is reachable from u in $\overleftarrow{D}$ if and only if u is reachable from v in D.

(*b*) By using part (ii) of (*a*), deduce from exercise 10.1.3*a* that if D is a digraph with no directed cycle, then $\delta^+ = 0$.

10.1.6 Show that if D is strict, then D contains a directed path of length at least $\max\{\delta^-, \delta^+\}$.

10.1.7 Show that if D is strict and $\max\{\delta^-, \delta^+\} = k > 0$, then D contains a directed cycle of length at least $k+1$.

10.1.8 Let $v_1, v_2, \ldots, v_\nu$ be the vertices of a digraph D. The *adjacency matrix* of D is the $\nu \times \nu$ matrix $\mathbf{A} = [a_{ij}]$ in which a_{ij} is the number of arcs of D with tail v_i and head v_j. Show that the (i, j)th entry of $\mathbf{A}^k$ is the number of directed (v_i, v_j)-walks of length k in D.

10.1.9 Let $D_1, D_2, \ldots, D_m$ be the dicomponents of D. The *condensation* $\hat{D}$ of D is a directed graph with m vertices $w_1, w_2, \ldots, w_m$; there is an arc in $\hat{D}$ with tail w_i and head w_j if and only if there is an arc in D with tail in D_i and head in D_j. Show that the condensation $\hat{D}$ of D contains no directed cycle.

10.1.10 Show that G has an orientation D such that $|d^+(v) - d^-(v)| \leq 1$ for all $v \in V$.

10.2 DIRECTED PATHS

There is no close relationship between the lengths of paths and directed paths in a digraph. That this is so is clear from the digraph of figure 10.3, which has no directed path of length greater than one.

FEUE WHEP

5753

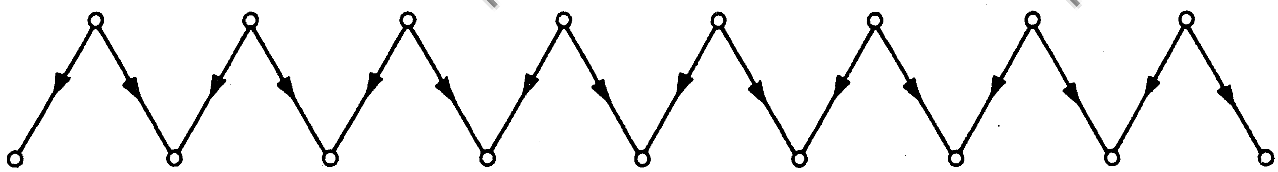


Figure 10.3

Surprisingly, some information about the lengths of directed paths in a digraph can be obtained by looking at its chromatic number. The following theorem, due to Roy (1967) and Gallai (1968), makes this precise.

Theorem 10.1 A digraph D contains a directed path of length $\chi - 1$.

Proof Let A' be a minimal set of arcs of D such that $D' = D - A'$ contains no directed cycle, and let the length of a longest directed path in D' be k. Now assign colours $1, 2, \ldots, k+1$ to the vertices of D' by assigning colour i to vertex v if the length of a longest directed path in D' with origin v is $i-1$. Denote by V_i the set of vertices with colour i. We shall show that $(V_1, V_2, \ldots, V_{k+1})$ is a proper $(k+1)$-vertex colouring of D.

First, observe that the origin and terminus of any directed path in D' have different colours. For let P be a directed (u, v)-path of positive length in D' and suppose $v \in V_i$. Then there is a directed path $Q = (v_1, v_2, \ldots, v_i)$ in D', where $v_1 = v$. Since D' contains no directed cycle, PQ is a directed path with origin u and length at least i. Thus $u \notin V_i$.

We can now show that the ends of any arc of D have different colours. Suppose $(u, v) \in A(D)$. If $(u, v) \in A(D')$ then (u, v) is a directed path in D' and so u and v have different colours. Otherwise, $(u, v) \in A'$. By the minimality of A', $D' + (u, v)$ contains a directed cycle C. $C - (u, v)$ is a directed (v, u)-path in D' and hence in this case, too, u and v have different colours.

Thus $(V_1, V_2, \ldots, V_{k+1})$ is a proper vertex colouring of D. It follows that $\chi \leq k+1$, and so D has a directed path of length $k \geq \chi - 1$ □

Theorem 10.1 is best possible in that every graph G has an orientation in which the longest directed path is of length $\chi - 1$. Given a proper χ-vertex colouring $(V_1, V_2, \ldots, V_\chi)$ of G, we orient G by converting edge uv to arc (u, v) if $u \in V_i$ and $v \in V_j$ with $i < j$. Clearly, no directed path in this orientation of G can contain more than χ vertices, since no two vertices of the path can have the same colour.

An orientation of a complete graph is called a *tournament*. The tournaments on four vertices are shown in figure 10.4. Each can be regarded as indicating the results of games in a round-robin tournament between four players; for example, the first tournament in figure 10.4 shows that one player has won all three games and that the other three have each won one.

A *directed Hamilton path* of D is a directed path that includes every

5754

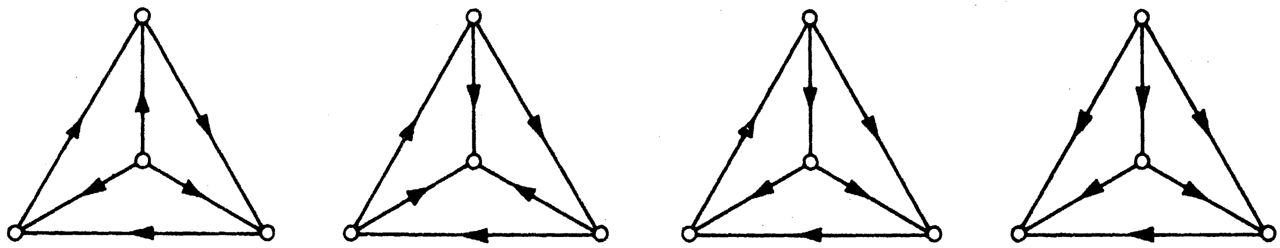


Figure 10.4. The tournaments on four vertices

vertex of D. An immediate corollary of theorem 10.1 is that every tournament has such a path. This was first proved by Rédei (1934).

Corollary 10.1 Every tournament has a directed Hamilton path.

Proof If D is a tournament, then $\chi = \nu$ □

Another interesting fact about tournaments is that there is always a vertex from which every other vertex can be reached in at most two steps. We shall obtain this as a special case of a theorem of Chvátal and Lovász (1974). An *in-neighbour* of a vertex v in D is a vertex u such that $(u, v) \in A$; an *out-neighbour* of v is a vertex w such that $(v, w) \in A$. We denote the sets of in-neighbours and out-neighbours of v in D by $N_D^-(v)$ and $N_D^+(v)$, respectively.

Theorem 10.2 A loopless digraph D has an independent set S such that each vertex of D not in S is reachable from a vertex in S by a directed path of length at most two.

Proof By induction on ν. The theorem holds trivially for $\nu = 1$. Assume that it is true for all digraphs with fewer than ν vertices, and let v be an arbitrary vertex of D. By the induction hypothesis there exists in $D' = D - (\{v\} \cup N^+(v))$ an independent set S' such that each vertex of D' not in S' is reachable from a vertex in S' by a directed path of length at most two. If v is an out-neighbour of some vertex u of S', then every vertex of $N^+(v)$ is reachable from u by a directed path of length two. Hence, in this case, $S = S'$ satisfies the required property. If, on the other hand, v is not an out-neighbour of any vertex of S', then v is joined to no vertex of S' and the independent set $S = S' \cup \{v\}$ has the required property □

Corollary 10.2 A tournament contains a vertex from which every other vertex is reachable by a directed path of length at most two.

Proof If D is a tournament, then $\alpha = 1$ □

5755

Exercises

10.2.1 Show that every tournament is either diconnected or can be transformed into a diconnected tournament by the reorientation of just one arc.

10.2.2* A digraph D is *unilateral* if, for any two vertices u and v, either v is reachable from u or u is reachable from v. Show that D is unilateral if and only if D has a spanning directed walk.

10.2.3 (*a*) Let $P = (v_1, v_2, \ldots, v_k)$ be a maximal directed path in a tournament D. Suppose that P is not a directed Hamilton path and let v be any vertex not on P. Show that, for some i, both (v_i, v) and (v, v_{i+1}) are arcs of D.
(*b*) Deduce Rédei's theorem.

10.2.4 Prove corollary 10.2 by considering a vertex of maximum outdegree.

10.2.5* (*a*) Let D be a digraph with $\chi > mn$, and let f be a real-valued function defined on V. Show that D has either a directed path $(u_0, u_1, \ldots, u_m)$ with $f(u_0) \leq f(u_1) \leq \ldots \leq f(u_m)$ or a directed path $(v_0, v_1, \ldots, v_n)$ with $f(v_0) > f(v_1) > \ldots > f(v_n)$.
(V. Chvátal and J. Komlós)
(*b*) Deduce that any sequence of $mn+1$ distinct integers contains either an increasing subsequence of m terms or a decreasing subsequence of n terms. (P. Erdös and G. Szekeres)

10.2.6 (*a*) Using theorem 10.1 and corollary 8.1.2, show that G has an orientation in which each directed path is of length at most Δ.
(*b*) Give a constructive proof of (*a*).

10.3 DIRECTED CYCLES

Corollary 10.1 tells us that every tournament contains a directed Hamilton path. Much stronger conclusions can be drawn, however, if the tournament is assumed to be diconnected. The following theorem is due to Moon (1966). If S and T are subsets of V, we denote by (S, T) the set of arcs of D that have their tails in S and their heads in T.

Theorem 10.3 Each vertex of a diconnected tournament D with $\nu \geq 3$ is contained in a directed k-cycle, $3 \leq k \leq \nu$.

Proof Let D be a diconnected tournament with $\nu \geq 3$, and let u be any vertex of D. Set $S = N^+(u)$ and $T = N^-(u)$. We first show that u is in a directed 3-cycle. Since D is diconnected, neither S nor T can be empty; and, for the same reason, (S, T) must be nonempty (see figure 10.5). There is thus some arc (v, w) in D with $v \in S$ and $w \in T$, and u is in the directed 3-cycle (u, v, w, u).

5756

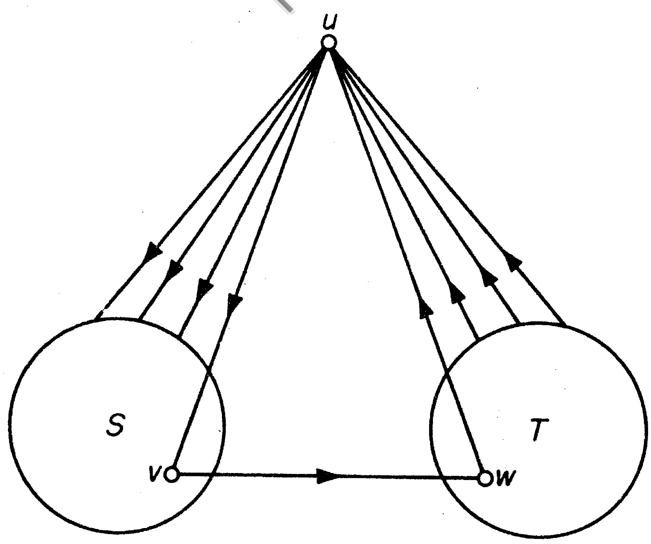


Figure 10.5

The theorem is now proved by induction on k. Suppose that u is in directed cycles of all lengths between 3 and n, where $n < \nu$. We shall show that u is in a directed $(n+1)$-cycle.

Let $C = (v_0, v_1, \ldots, v_n)$ be a directed n-cycle in which $v_0 = v_n = u$. If there is a vertex v in $V(D)\backslash V(C)$ which is both the head of an arc with tail in C and the tail of an arc with head in C, then there are adjacent vertices v_i and v_{i+1} on C such that both (v_i, v) and (v, v_{i+1}) are arcs of D. In this case u is in the directed $(n+1)$-cycle $(v_0, v_1, \ldots, v_i, v, v_{i+1}, \ldots, v_n)$.

Otherwise, denote by S the set of vertices in $V(D)\backslash V(C)$ which are heads of arcs joined to C, and by T the set of vertices in $V(D)\backslash V(C)$ which are tails of arcs joined to C (see figure 10.6).

As before, since D is diconnected, S, T and (S, T) are all nonempty, and there is some arc (v, w) in D with $v \in S$ and $w \in T$. Hence u is in the directed $(n+1)$-cycle $(v_0, v, w, v_2, \ldots, v_n)$ □

A *directed Hamilton cycle* of D is a directed cycle that includes every vertex of D. It follows from theorem 10.3 (and was first proved by Camion, 1959) that every diconnected tournament contains such a cycle. The next

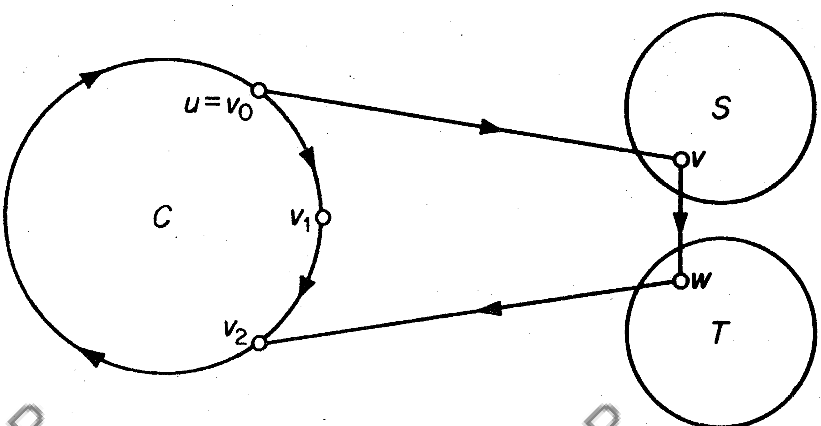


Figure 10.6

FEUE WHEP

5757

theorem extends Dirac's theorem (4.3) to digraphs. It is a special case of a theorem due to Ghouila-Houri (1960).

Theorem 10.4 If D is strict and $\min\{\delta^-, \delta^+\} \geq \nu/2 > 1$, then D contains a directed Hamilton cycle.

Proof Suppose that D satisfies the hypotheses of the theorem, but does not contain a directed Hamilton cycle. Denote the length of a longest directed cycle in D by l, and let $C = (v_1, v_2, \ldots, v_l, v_1)$ be a directed cycle in D of length l. We note that $l > \nu/2$ (exercise 10.1.7). Let P be a longest directed path in $D - V(C)$ and suppose that P has origin u, terminus v and length m (see figure 10.7). Clearly

$$\nu \geq l + m + 1 \tag{10.1}$$

and, since $l > \nu/2$,

$$m < \nu/2 \tag{10.2}$$

Set

$$S = \{i \mid (v_{i-1}, u) \in A\} \quad \text{and} \quad T = \{i \mid (v, v_i) \in A\}$$

We first show that S and T are disjoint. Let $C_{j,k}$ denote the section of C with origin v_j and terminus v_k. If some integer i were in both S and T, D would contain the directed cycle $C_{i,i-1}(v_{i-1}, u)P(v, v_i)$ of length $l + m + 1$, contradicting the choice of C. Thus

$$S \cap T = \emptyset \tag{10.3}$$

Now, because P is a maximal directed path in $D - V(C)$, $N^-(u) \subseteq V(P) \cup V(C)$. But the number of in-neighbours of u in C is precisely $|S|$ and so $d_D^-(u) = d_P^-(u) + |S|$. Since $d_D^-(u) \geq \delta^- \geq \nu/2$ and $d_P^-(u) \leq m$,

$$|S| \geq \nu/2 - m \tag{10.4}$$

A similar argument yields

$$|T| \geq \nu/2 - m \tag{10.5}$$

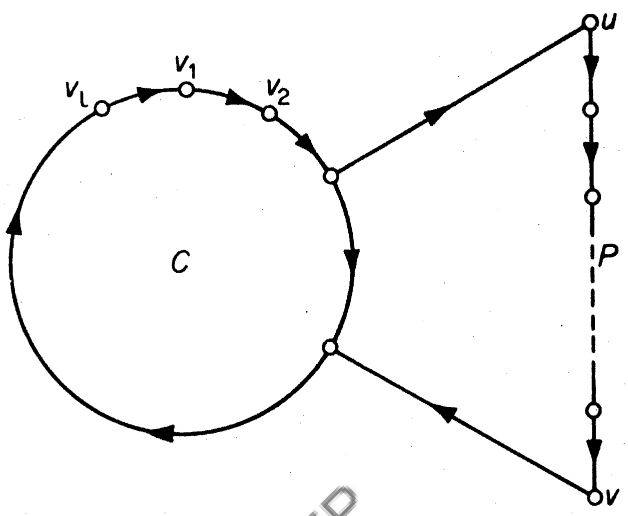


Figure 10.7

5758

Note that, by (10.2), both S and T are nonempty. Adding (10.4) and (10.5) and using (10.1), we obtain

$$|S|+|T| \geq l-m+1$$

and therefore, by (10.3),

$$|S \cup T| \geq l-m+1 \tag{10.6}$$

Since S and T are disjoint and nonempty, there are positive integers i and k such that $i \in S$, $i+k \in T$ and

$$i+j \notin S \cup T \quad \text{for} \quad 1 \leq j < k \tag{10.7}$$

where addition is taken modulo l.

From (10.6) and (10.7) we see that $k \leq m$. Thus the directed cycle $C_{i+k,i-1}(v_{i-1}, u)P(v, v_{i+k})$, which has length $l+m+1-k$, is longer than C. This contradiction establishes the theorem □

Exercises

10.3.1 Show how theorem 4.3 can be deduced from theorem 10.4.

10.3.2 A *directed Euler tour* of D is a directed tour that traverses each arc of D exactly once. Show that D contains a directed Euler tour if and only if D is connected and $d^+(v) = d^-(v)$ for all $v \in V$.

10.3.3 Let D be a digraph such that

(i) $d^+(x) - d^-(x) = l = d^-(y) - d^+(y)$;
(ii) $d^+(v) = d^-(v)$ for $v \in V \backslash \{x, y\}$.

Show, using exercise 10.3.2, that there exist l arc-disjoint directed (x, y)-paths in D.

10.3.4* Show that a diconnected digraph which contains an odd cycle, also contains a directed odd cycle.

10.3.5 A nontrivial digraph D is *k-arc-connected* if, for every nonempty proper subset S of V, $|(S, \bar{S})| \geq k$. Show that a nontrivial digraph is diconnected if and only if it is 1-arc-connected.

10.3.6 The *associated digraph* $D(G)$ of a graph G is the digraph obtained when each edge e of G is replaced by two oppositely oriented arcs with the same ends as e. Show that

(*a*) there is a one-one correspondence between paths in G and directed paths in $D(G)$;
(*b*) $D(G)$ is k-arc-connected if and only if G is k-edge-connected.

APPLICATIONS

10.4 A JOB SEQUENCING PROBLEM

A number of jobs $J_1, J_2, \ldots, J_n$, have to be processed on one machine; for example, each J_i might be an order of bottles or jars in a glass factory. After

FEUE WHEP

5759

each job, the machine must be adjusted to fit the requirements of the next job. If the time of adaptation from job J_i to job J_j is t_{ij}, find a sequencing of the jobs that minimises the total machine adjustment time.

This problem is clearly related to the travelling salesman problem, and no efficient method for its solution is known. It is therefore desirable to have a method for obtaining a reasonably good (but not necessarily optimal) solution. Our method makes use of Rédei's theorem (corollary 10.1).

Step 1 Construct a digraph D with vertices $v_1, v_2, \ldots, v_n$, such that $(v_i, v_j) \in A$ if and only if $t_{ij} \le t_{ji}$. By definition, D contains a spanning tournament.

Step 2 Find a directed Hamilton path $(v_{i_1}, v_{i_2}, \ldots, v_{i_n})$ of D (exercise 10.4.1), and sequence the jobs accordingly.

Since step 1 discards the larger half of the adjustment matrix $[t_{ij}]$, it is a reasonable supposition that this method, in general, produces a fairly good job sequence. Note, however, that when the adjustment matrix is symmetric, the method is of no help whatsoever.

As an example, suppose that there are six jobs J_1, J_2, J_3, J_4, J_5 and J_6 and that the adjustment matrix is

	J_1	J_2	J_3	J_4	J_5	J_6
J_1	0	5	3	4	2	1
J_2	1	0	1	2	3	2
J_3	2	5	0	1	2	3
J_4	1	4	4	0	1	2
J_5	1	3	4	5	0	5
J_6	4	4	2	3	1	0

The sequence $J_1 \to J_2 \to J_3 \to J_4 \to J_5 \to J_6$ requires 13 units in adjustment time. To find a better sequence, construct the digraph D as in step 1 (figure 10.8).

$(v_1, v_6, v_3, v_4, v_5, v_2)$ is a directed Hamilton path of D, and yields the sequence

$$J_1 \to J_6 \to J_3 \to J_4 \to J_5 \to J_2$$

which requires only eight units of adjustment time. Note that the reverse sequence

$$J_2 \to J_5 \to J_4 \to J_3 \to J_6 \to J_1$$

is far worse, requiring 19 units of adjustment time.

Exercises

10.4.1 With the aid of exercise 10.2.3, describe a good algorithm for finding a directed Hamilton path in a tournament.

FEUE WHEP

5760

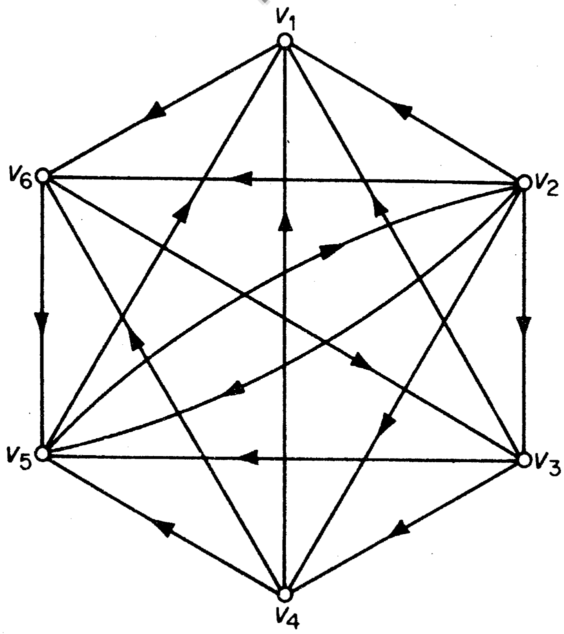


Figure 10.8

10.4.2 Show, by means of an example, that a sequencing of jobs obtained by the above method may be far from optimal.

10.5 DESIGNING AN EFFICIENT COMPUTER DRUM

The position of a rotating drum is to be recognised by means of binary signals produced at a number of electrical contacts at the surface of the drum. The surface is divided into 2^n sections, each consisting of either insulating or conducting material. An insulated section gives signal 0 (no current), whereas a conducting section gives signal 1 (current). For example, the position of the drum in figure 10.9 gives a reading 0010 at the four

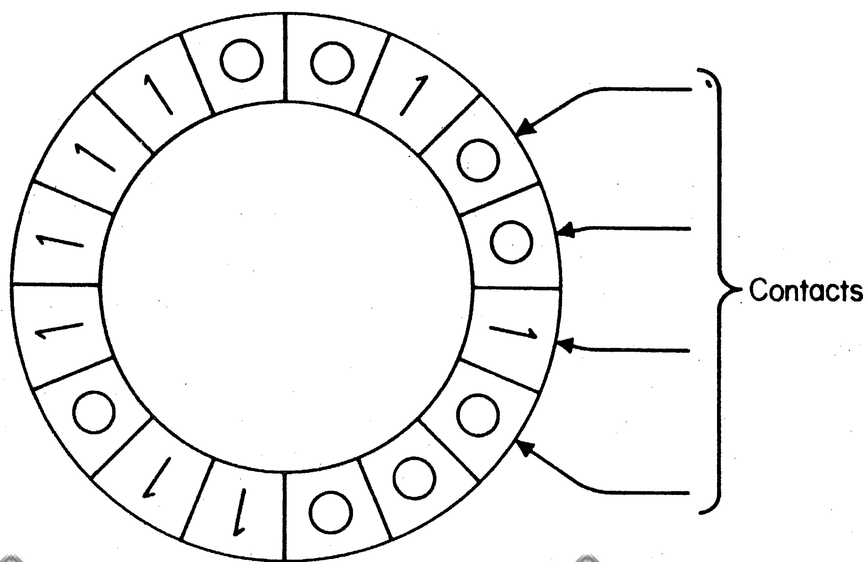


Figure 10.9. A computer drum

FEUE WHEP

5761

contacts. If the drum were rotated clockwise one section, the reading would be 1001. Thus these two positions can be distinguished, since they give different readings. However, a further rotation of two sections would result in another position with reading 0010, and therefore this latter position is indistinguishable from the initial one.

We wish to design the drum surface in such a way that the 2^n different positions of the drum can be distinguished by k contacts placed consecutively around part of the drum, and we would like this number k to be as small as possible. How can this be accomplished?

First note that k contacts yield a k-digit binary number, and there are 2^k such numbers. Therefore, if all 2^n positions are to give different readings, we must have $2^k \geq 2^n$, that is, $k \geq n$. We shall show that the surface of the drum can be designed in such a way that n contacts suffice to distinguish all 2^n positions.

We define a digraph D_n as follows: the vertices of D_n are the $(n-1)$-digit binary numbers $p_1p_2 \ldots p_{n-1}$ with $p_i = 0$ or 1. There is an arc with tail $p_1p_2 \ldots p_{n-1}$ and head $q_1q_2 \ldots q_{n-1}$ if and only if $p_{i+1} = q_i$ for $1 \leq i \leq n-2$; in other words, all arcs are of the form $(p_1p_2 \ldots p_{n-1}, p_2p_3 \ldots p_n)$. In addition, each arc $(p_1p_2 \ldots p_{n-1}, p_2p_3 \ldots p_n)$ of D_n is assigned the label $p_1p_2 \ldots p_n$. D_4 is shown in figure 10.10.

Clearly, D_n is connected and each vertex of D_n has indegree two and outdegree two. Therefore (exercise 10.3.2) D_n has a directed Euler tour. This directed Euler tour, regarded as a sequence of arcs of D_n, yields a binary sequence of length 2^n suitable for the design of the drum surface.

For example, the digraph D_4 of figure 10.10 has a directed Euler tour $(a_1, a_2, \ldots, a_{16})$, giving the 16-digit binary sequence 0000111100101101. (Just read off the first digits of the labels of the a_i.) A drum constructed from this sequence is shown in figure 10.11.

This application of directed Euler tours is due to Good (1946).

Exercises

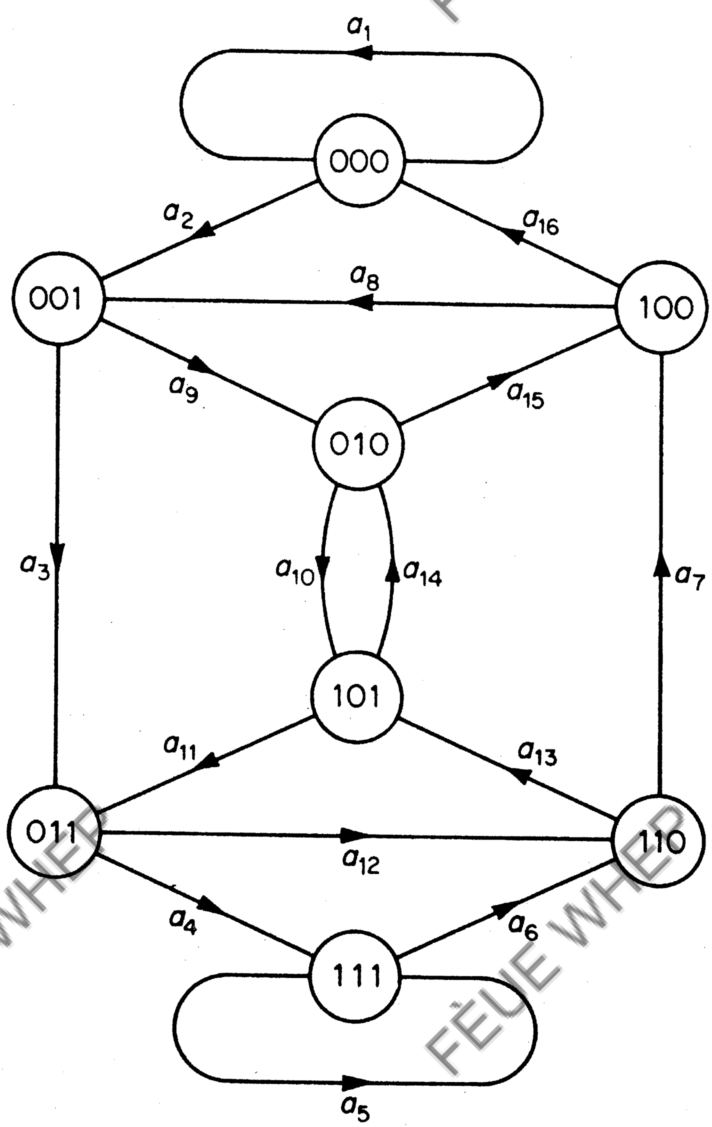
10.5.1. Find a circular sequence of seven 0's and seven 1's such that all 4-digit binary numbers except 0000 and 1111 appear as blocks of the sequence.

10.5.2 Let S be an alphabet of n letters. Show that there is a circular sequence containing n^3 copies of each letter such that every four-letter 'word' formed from letters of S appears as a block of the sequence.

10.6 MAKING A ROAD SYSTEM ONE-WAY

Given a road system, how can it be converted to one-way operation so that traffic may flow as smoothly as possible?

5762



Arc	Label
a_1	0000
a_2	0001
a_3	0011
a_4	0111
a_5	1111
a_6	1110
a_7	1100
a_8	1001
a_9	0010
a_{10}	0101
a_{11}	1011
a_{12}	0110
a_{13}	1101
a_{14}	1010
a_{15}	0100
a_{16}	1000

Figure 10.10

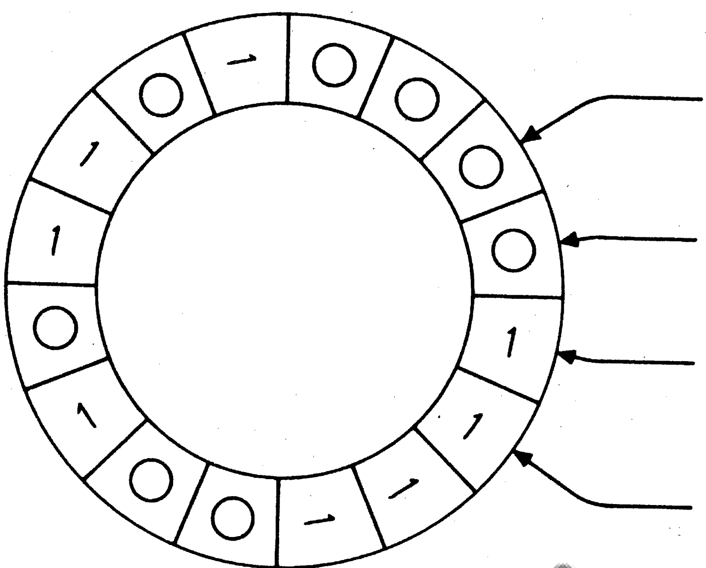


Figure 10.11

5763

This is clearly a problem on orientations of graphs. Consider, for example, the two graphs, representing road networks, in figures 10.12*a* and 10.12*b*. No matter how G_1 may be oriented, the resulting orientation cannot be diconnected—traffic will not be able to flow freely through the system. The trouble is that G_1 has a cut edge. On the other hand G_2 has the 'balanced' orientation D_2 (figure 10.12*c*), in which each vertex is reachable from each other vertex in at most two steps; in particular D_2 is diconnected.

Certainly, a necessary condition for G to have a diconnected orientation is that G be 2-edge-connected. Robbins (1939) showed that this condition is also sufficient.

Theorem 10.5 If G is 2-edge-connected, then G has a diconnected orientation.

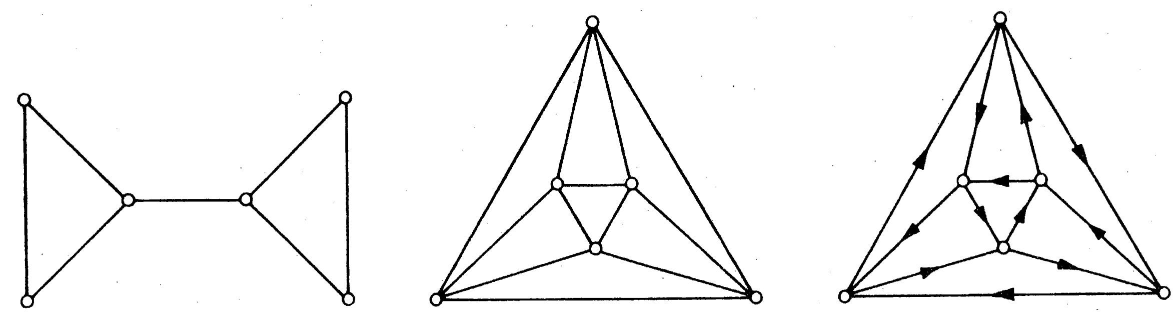
Proof Let G be 2-edge-connected. Then G contains a cycle G_1. We define inductively a sequence $G_1, G_2, \ldots$ of connected subgraphs of G as follows: if G_i $(i = 1, 2, \ldots)$ is not a spanning subgraph of G, let v_i be a vertex of G not in G_i. Then (exercise 3.2.1) there exist edge-disjoint paths P_i and Q_i from v_i to G_i. Define

$$G_{i+1} = G_i \cup P_i \cup Q_i$$

Since $\nu(G_{i+1}) > \nu(G_i)$, this sequence must terminate in a spanning subgraph G_n of G.

We now orient G_n by orienting G_1 as a directed cycle, each path P_i as a directed path with origin v_i, and each path Q_i as a directed path with terminus v_i. Clearly every G_i, and hence in particular G_n, is thereby given a diconnected orientation. Since G_n is a spanning subgraph of G it follows that G, too, has a diconnected orientation □

Nash-Williams (1960) has generalised Robbins' theorem by showing that every $2k$-edge-connected graph G has a k-arc-connected orientation. Although the proof of this theorem is difficult, the special case when G has an Euler trail admits of a simple proof.



(a) (b) (c)

Figure 10.12. (*a*) G_1; (*b*) G_2; (*c*) D_2

5764

Theorem 10.6 Let G be a $2k$-edge-connected graph with an Euler trail. Then G has a k-arc-connected orientation.

Proof Let $v_0e_1v_1 \ldots e_\varepsilon v_\varepsilon$ be an Euler trail of G. Orient G by converting the edge e_i with ends v_{i-1} and v_i to an arc a_i with tail v_{i-1} and head v_i, for $1 \le i \le \varepsilon$. Now let $[S, \bar{S}]$ be an m-edge cut of G. The number of times the directed trail $(v_0, a_1, v_1, \ldots, a_\varepsilon, v_\varepsilon)$ crosses from S to $\bar{S}$ differs from the number of times it crosses from $\bar{S}$ to S by at most one. Since it includes all arcs of D, both $(S, \bar{S})$ and $(\bar{S}, S)$ must contain at least $[m/2]$ arcs. The result follows □

Exercises

10.6.1 Show, by considering the Petersen graph, that the following statement is false: every graph G has an orientation in which, for every $S \subseteq V$, the cardinalities of $(S, \bar{S})$ and $(\bar{S}, S)$ differ by at most one.

10.6.2 (*a*) Show that Nash-Williams' theorem is equivalent to the following statement: if every bond of G has at least $2k$ edges, then there is an orientation of G in which every bond has at least k arcs in each direction.

(*b*) Show, by considering the Grötzsch graph (figure 8.2), that the following analogue of Nash-Williams' theorem is false: if every cycle of G has at least $2k$ edges, then there is an orientation of G in which every cycle has at least k arcs in each direction.

10.7 RANKING THE PARTICIPANTS IN A TOURNAMENT

A number of players each play one another in a tennis tournament. Given the outcomes of the games, how should the participants be ranked?

Consider, for example, the tournament of figure 10.13. This represents the result of a tournament between six players; we see that player 1 beat players 2, 4, 5 and 6 and lost to player 3, and so on.

One possible approach to ranking the participants would be to find a directed Hamilton path in the tournament (such a path exists by virtue of corollary 10.1), and then rank according to the position on the path. For instance, the directed Hamilton path (3, 1, 2, 4, 5, 6) would declare player 3 the winner, player 1 runner-up, and so on. This method of ranking, however, does not bear further examination, since a tournament generally has many directed Hamilton paths; our example has (1, 2, 4, 5, 6, 3), (1, 4, 6, 3, 2, 5) and several others.

Another approach would be to compute the scores (numbers of games won by each player) and compare them. If we do this we obtain the score vector

$$\mathbf{s}_1 = (4, 3, 3, 2, 2, 1)$$

5765

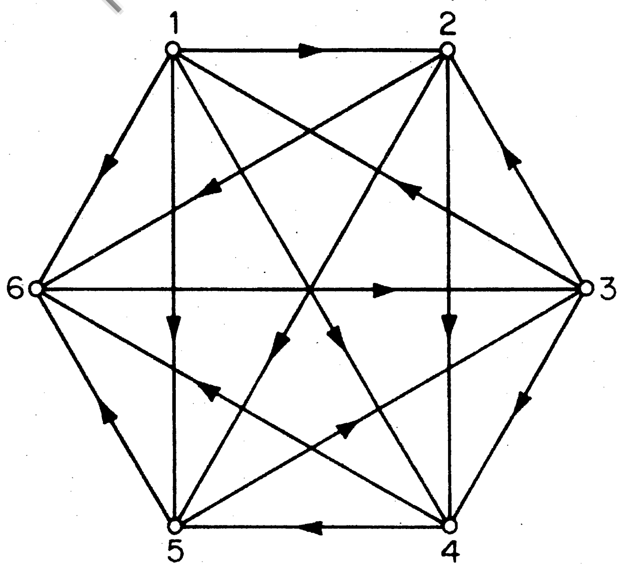


Figure 10.13

The drawback here is that this score vector does not distinguish between players 2 and 3 even though player 3 beat players with higher scores than did player 2. We are thus led to the second-level score vector

$$\mathbf{s}_2 = (8, 5, 9, 3, 4, 3)$$

in which each player's second-level score is the sum of the scores of the players he beat. Player 3 now ranks first. Continuing this procedure we obtain further vectors

$$\mathbf{s}_3 = (15, 10, 16, 7, 12, 9)$$
$$\mathbf{s}_4 = (38, 28, 32, 21, 25, 16)$$
$$\mathbf{s}_5 = (90, 62, 87, 41, 48, 32)$$
$$\mathbf{s}_6 = (183, 121, 193, 80, 119, 87)$$

The ranking of the players is seen to fluctuate a little, player 3 vying with player 1 for first place. We shall show that this procedure always converges to a fixed ranking when the tournament in question is diconnected and has at least four vertices. This will then lead to a method of ranking the players in any tournament.

In a diconnected digraph D, the length of a shortest directed (u, v)-path is denoted by $\vec{d}_D(u, v)$ and is called the *distance from u to v*; the *directed diameter* of D is the maximum distance from any one vertex of D to any other.

Theorem 10.7 Let D be a diconnected tournament with $\nu \geq 5$, and let $\mathbf{A}$ be the adjacency matrix of D. Then $\mathbf{A}^{d+3} > \mathbf{0}$ (every entry positive), where d is the directed diameter of D.

5766

Proof The (i, j)th entry of $\mathbf{A}^k$ is precisely the number of directed (v_i, v_j)-walks of length k in D (exercise 10.1.8). We must therefore show that, for any two vertices v_i and v_j (possibly identical), there is a directed (v_i, v_j)-walk of length $d+3$.

Let $d_{ij} = \vec{d}(v_i, v_j)$. Then $0 \le d_{ij} \le d \le \nu - 1$ and therefore

$$3 \le d - d_{ij} + 3 \le \nu + 2$$

If $d - d_{ij} + 3 \le \nu$ then, by theorem 10.3, there is a directed $(d - d_{ij} + 3)$-cycle C containing v_j. A directed (v_i, v_j)-path P of length d_{ij} followed by the directed cycle C together form a directed (v_i, v_j)-walk of length $d+3$, as desired.

There are two special cases. If $d - d_{ij} + 3 = \nu + 1$, then P followed by a directed $(\nu-2)$-cycle through v_j followed by a directed 3-cycle through v_j constitute a directed (v_i, v_j)-walk of length $d+3$ (the $(\nu-2)$-cycle exists since $\nu \ge 5$); and if $d - d_{ij} + 3 = \nu + 2$, then P followed by a directed $(\nu-1)$-cycle through v_j followed by a directed 3-cycle through v_j constitute such a walk □

A real matrix $\mathbf{R}$ is called *primitive* if $\mathbf{R}^k > \mathbf{0}$ for some k.

Corollary 10.7 The adjacency matrix $\mathbf{A}$ of a tournament D is primitive if and only if D is diconnected and $\nu \ge 4$.

Proof If D is not diconnected, then there are vertices v_i and v_j in D such that v_j is not reachable from v_i. Thus there is no directed (v_i, v_j)-walk in D. It follows that the (i, j)th entry of $\mathbf{A}^k$ is zero for all k, and hence $\mathbf{A}$ is not primitive.

Conversely, suppose that D is diconnected. If $\nu \ge 5$ then, by theorem 10.7, $\mathbf{A}^{d+3} > \mathbf{0}$ and so $\mathbf{A}$ is primitive. There is just one diconnected tournament on three vertices (figure 10.14a), and just one diconnected tournament on four vertices (figure 10.14b). It is readily checked that the adjacency

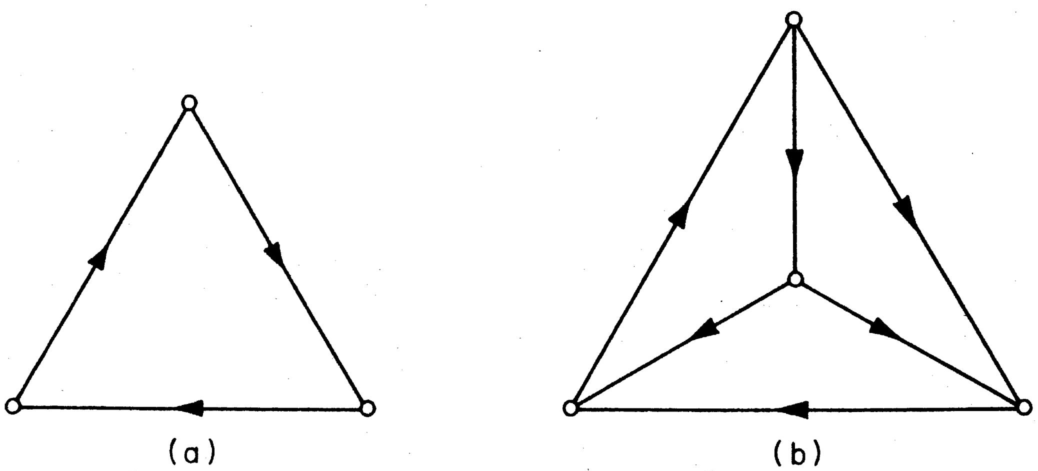


Figure 10.14

5767

matrix of the 3-vertex tournament is not primitive, and it can be shown that the ninth power of the adjacency matrix of the 4-vertex tournament has all entries positive □

Returning now to the score vectors, we see that the ith-level score vector in a tournament D is given by

$$\mathbf{s}_i = \mathbf{A}^i\mathbf{J}$$

where $\mathbf{A}$ is the adjacency matrix of D, and $\mathbf{J}$ is a column vector of 1's. If the matrix $\mathbf{A}$ is primitive then, by the Perron–Frobenius theorem (see Gantmacher, 1960), the eigenvalue of $\mathbf{A}$ with largest absolute value is a real positive number r and, furthermore,

$$\lim_{i\to\infty}\left(\frac{\mathbf{A}}{r}\right)^i\mathbf{J} = \mathbf{s}$$

where $\mathbf{s}$ is a positive eigenvector of $\mathbf{A}$ corresponding to r. Therefore, by corollary 10.7, if D is a diconnected tournament on at least four vertices, the normalised vector $\bar{\mathbf{s}}$ (with entries summing to one) can be taken as the vector of relative strengths of the players in D. In the example of figure 10.13, we find that (approximately)

$$r = 2.232 \quad \text{and} \quad \bar{\mathbf{s}} = (.238, .164, .231, .113, .150, .104)$$

Thus the ranking of the players given by this method is 1, 3, 2, 5, 4, 6.

If the tournament is not diconnected, then (exercises 10.1.9 and 10.1.3b) its dicomponents can be linearly ordered so that the ordering preserves dominance. The participants in a round-robin tournament can now be ranked according to the following procedure.

Step 1 In each dicomponent on four or more vertices, rank the players using the eigenvector $\bar{\mathbf{s}}$; in a dicomponent on three vertices rank all three players equal.

Step 2 Rank the dicomponents in their dominance-preserving linear order $D_1, D_2, \ldots, D_m$; that is, if $i<j$ then every arc with one end in D_i and one end in D_j has its head in D_j.

This method of ranking is due to Wei (1952) and Kendall (1955). For other ranking procedures, see Moon and Pullman (1970).

Exercises

10.7.1 Apply the method of ranking described in section 10.7 to

(*a*) the four tournaments shown in figure 10.4;

(*b*) the tournament with adjacency matrix

5768

	A	B	C	D	E	F	G	H	I	J
A	0	1	1	1	1	1	0	0	1	1
B	0	0	1	0	0	1	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	1	1	0	1	1	0	0	1	0
E	0	1	1	0	0	0	0	0	0	0
F	0	0	1	0	1	0	0	0	0	0
G	1	1	1	1	1	1	0	0	1	0
H	1	1	1	1	1	1	1	0	1	1
I	0	1	1	0	1	0	0	0	0	0
J	0	1	1	1	1	1	1	0	1	0

10.7.2 An alternative method of ranking is to consider 'loss vectors' instead of score vectors.

(*a*) Show that this amounts to ranking the converse tournament and then reversing the ranking so found.

(*b*) By considering the diconnected tournament on four vertices, show that the two methods of ranking do not necessarily yield the same result.

REFERENCES

Camion, P. (1959). Chemins et circuits hamiltoniens des graphes complets. *C.R. Acad. Sci. Paris*, **249,** 2151–52

Chvátal, V. and Lovász, L. (1974). Every directed graph has a semi-kernel, in *Hypergraph Seminar* (eds. C. Berge and D. K. Ray-Chaudhuri), Springer-Verlag, New York, p. 175

Gallai, T. (1968). On directed paths and circuits, in *Theory of Graphs* (eds. P. Erdös and G. Katona), Academic Press, New York, pp. 115–18

Gantmacher F. R. (1960). *Theory of Matrices*, Chelsea, New York

Ghouila-Houri, A. (1960). Une condition suffisante d'existence d'un circuit hamiltonien. *C. R. Acad. Sci. Paris*, **251,** 495–97

Good, I. J. (1946). Normal recurring decimals. *J. London Math. Soc.* **21,** 167–72

Kendall, M. G. (1955). Further contributions to the theory of paired comparisons. *Biometrics*, **11,** 43–62

Moon, J. W. (1966). On subtournaments of a tournament. *Canad. Math. Bull.*, **9,** 297–301

Moon, J. W. and Pullman, N. J. (1970). On generalised tournament matrices. *SIAM Rev*, **12,** 384–399

Nash-Williams, C. St. J. A. (1960). On orientations, connectivity and odd-vertex pairings in finite graphs. *Canad. J. Math.*, **12,** 555–67

5769

Rédei, L. (1934). Ein kombinatorischer Satz. *Acta Litt. Sci. Szeged*, **7,** 39–43

Robbins, H. E. (1939). A theorem on graphs, with an application to a problem of traffic control. *Amer. Math. Monthly*, **46,** 281–83

Roy, B. (1967). Nombre chromatique et plus longs chemins d'un graphe, *Rev. Francaise Automat. Informat. Recherche Opérationelle Sér. Rouge*, **1,** 127–32

Wei, T. H. (1952). *The Algebraic Foundations of Ranking Theory*. Ph.D. Thesis, Cambridge University

FĒUE WHEP

5770

FEUE WHEP

11 Networks

11.1 FLOWS

Transportation networks, the means by which commodities are shipped from their production centres to their markets, can be most effectively analysed when they are viewed as digraphs that possess some additional structure. The resulting theory is the subject of this chapter. It has a wide range of important applications.

A *network* N is a digraph D (the *underlying digraph* of N) with two distinguished subsets of vertices, X and Y, and a non-negative integer-valued function c defined on its arc set A; the sets X and Y are assumed to be disjoint and nonempty. The vertices in X are the *sources* of N and those in Y are the *sinks* of N. They correspond to production centres and markets, respectively. Vertices which are neither sources nor sinks are called *intermediate vertices;* the set of such vertices will be denoted by I. The function c is the *capacity function* of N and its value on an arc a the *capacity* of a. The capacity of an arc can be thought of as representing the maximum rate at which a commodity can be transported along it.

We represent a network by drawing its underlying digraph and labelling each arc with its capacity. Figure 11.1 shows a network with two sources x_1 and x_2, three sinks y_1, y_2 and y_3, and four intermediate vertices v_1, v_2, v_3 and v_4.

If $S \subseteq V$, we denote $V \backslash S$ by $\bar{S}$. In addition, we shall find the following notation useful. If f is a real-valued function defined on the arc set A of N, and if $K \subseteq A$, we denote $\sum_{a \in K} f(a)$ by $f(K)$. Furthermore, if K is a set of arcs of the form $(S, \bar{S})$, we shall write $f^+(S)$ for $f(S, \bar{S})$ and $f^-(S)$ for $f(\bar{S}, S)$.

A *flow* in a network N is an integer-valued function f defined on A such that

$$0 \leq f(a) \leq c(a) \quad \text{for all} \quad a \in A \tag{11.1}$$

and

$$f^-(v) = f^+(v) \quad \text{for all} \quad v \in I \tag{11.2}$$

The value $f(a)$ of f on an arc a can be likened to the rate at which material is transported along a under the flow f. The upper bound in condition (11.1) is called the *capacity constraint;* it imposes the natural restriction that the rate of flow along an arc cannot exceed the capacity of the arc. Condition (11.2), called the *conservation condition*, requires that, for any intermediate vertex v, the rate at which material is transported into v is

5771

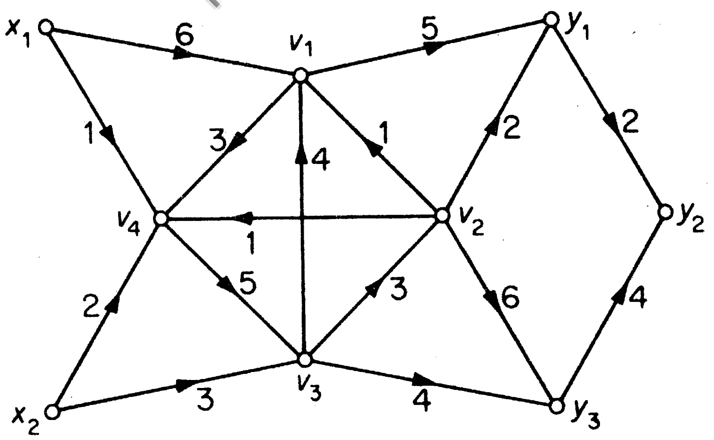


Figure 11.1. A network

equal to the rate at which it is transported out of v. Note that every network has at least one flow, since the function f defined by $f(a)=0$, for all $a \in A$, clearly satisfies both (11.1) and (11.2); it is called the *zero flow*. A less trivial example of a flow is given in figure 11.2. The flow along each arc is indicated in bold type.

If S is a subset of vertices in a network N and f is a flow in N, then $f^+(S)-f^-(S)$ is called the *resultant flow out of* S, and $f^-(S)-f^+(S)$ the *resultant flow into* S, relative to f. Since the conservation condition requires that the resultant flow out of any intermediate vertex is zero, it is intuitively clear and not difficult to show (exercise 11.1.3) that, relative to any flow f, the resultant flow out of X is equal to the resultant flow into Y. This common quantity is called the *value* of f, and is denoted by val f; thus

$$\operatorname{val} f = f^+(X) - f^-(X)$$

The value of the flow indicated in figure 11.2 is 6.

A flow f in N is a *maximum flow* if there is no flow f' in N such that $\operatorname{val} f' > \operatorname{val} f$. Such flows are of obvious importance in the context of transportation networks. The problem of determining a maximum flow in an arbitrary network can be reduced to the case of networks that have just one

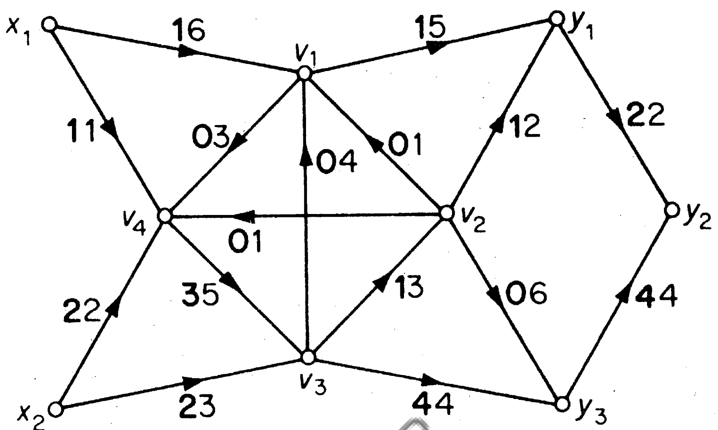


Figure 11.2. A flow in a network

5772

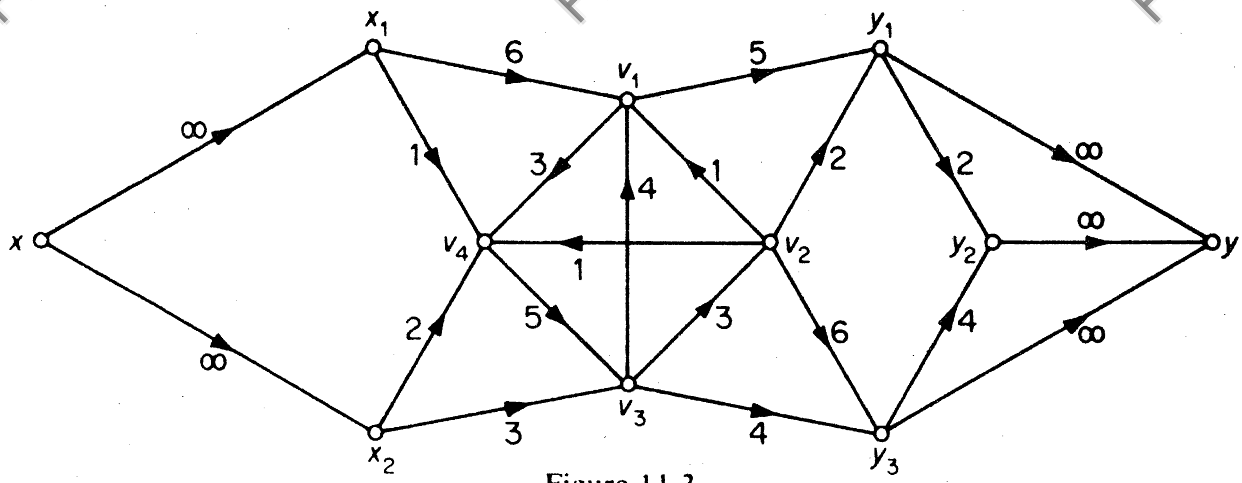


Figure 11.3

source and one sink by means of a simple device. Given a network N, construct a new network N' as follows:

(i) adjoin two new vertices x and y to N;
(ii) join x to each vertex in X by an arc of capacity ∞;
(iii) join each vertex in Y to y by an arc of capacity ∞;
(iv) designate x as the source and y as the sink of N'.

Figure 11.3 illustrates this procedure as applied to the network N of figure 11.1.

Flows in N and N' correspond to one another in a simple way. If f is a flow in N such that the resultant flow out of each source and into each sink is non-negative (it suffices to restrict our attention to such flows) then the function f' defined by

$$f'(a)=\begin{cases} f(a) & \text{if} \quad a \text{ is an arc of } N \\ f^+(v)-f^-(v) & \text{if} \quad a=(x,v) \\ f^-(v)-f^+(v) & \text{if} \quad a=(v,y)\end{cases} \tag{11.3}$$

is a flow in N' such that $\operatorname{val} f' = \operatorname{val} f$ (exercise 11.1.4*a*). Conversely, the restriction to the arc set of N of a flow in N' is a flow in N having the same value (exercise 11.1.4*b*). Therefore, throughout the next three sections, we shall confine our attention to networks that have a single source x and a single sink y.

Exercises

11.1.1 For each of the following networks (see diagram, p. 194), determine all possible flows and the value of a maximum flow.

11.1.2 Show that, for any flow f in N and any $S \subseteq V$,

$$\sum_{v\in S} (f^+(v)-f^-(v)) = f^+(S)-f^-(S)$$

(Note that, in general, $\sum_{v\in S} f^+(v) \neq f^+(S)$ and $\sum_{v\in S} f^-(v) \neq f^-(S)$).