DE GRUYTER

Ulrich Knauer

ALGEBRAIC GRAPH THEORY

MORPHISMS, MONOIDS AND MATRICES

STUDIES IN MATHEMATICS 41

De Gruyter Studies in Mathematics 41

Ulrich Knauer

Algebraic Graph Theory

Morphisms, Monoids and Matrices

De Gruyter

Mathematical Subject Classification 2010: 05C05, 05C10, 05C12, 05C20, 05C25, 05C38, 05C50, 05C62, 05C75, 05C90, 20M17, 20M19, 20M20, 20M30, 18B10, 18B40.

ISBN 978-3-11-025408-2
e-ISBN 978-3-11-025509-6
ISSN 0179-0986

Library of Congress Cataloging-in-Publication Data

Knauer, U., 1942–
Algebraic graph theory : morphisms, monoids, and matrices / by Ulrich Knauer.
p. cm. – (De Gruyter studies in mathematics ; 41)
Includes bibliographical references and index.
ISBN 978-3-11-025408-2 (alk. paper)
1. Graph theory. 2. Algebraic topology. I. Title.
QA166.K53 2011
512'.5–dc23
2011017050

Bibliographic information published by the Deutsche Nationalbibliothek

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

Typesetting: Da-TeX Gerd Blumenstein, Leipzig, www.da-tex.de
Printing and binding: Hubert & Co. GmbH & Co. KG, Göttingen
♾ Printed on acid-free paper

Printed in Germany

www.degruyter.com

Preface

This book is a collection of the lectures I have given on algebraic graph theory. These lectures were designed for mathematics students in a Master's program, but they may also be of interest to undergraduates in the final year of a Bachelor's curriculum.

The lectures cover topics which can be used as starting points for a Master's or Bachelor's thesis. Some questions raised in the text could even be suitable as subjects of doctoral dissertations. The advantage afforded by the field of algebraic graph theory is that it allows many questions to be understood from a general mathematical background and tackled almost immediately.

In fact, my lectures have also been attended by graduate students in informatics with a minor in mathematics. In computer science and informatics, many of the concepts associated with graphs play an important role as structuring tools – they enable us to model a wide variety of different systems, such as the structure of physical networks (of roads, computers, telephones etc.) as well as abstract data structures (e.g. lists, stacks, trees); functional and object oriented programming are also based on graphs as a means of describing discrete entities. In addition, category theory is gaining more and more importance in informatics; therefore, these lectures also include a basic and concrete introduction to categories, with numerous examples and applications.

I gave the lectures first at the University of Bielefeld and then, in various incarnations, at the Carl von Ossietzky Universität Oldenburg. They were sometimes presented in English and in several other countries, including Thailand and New Zealand.

Selection of topics

The choice of topics is in part standard, but it also reflects my personal preferences. Many students seem to have found the chosen topics engaging, as well as helpful and useful in getting started on thesis research at various levels.

To mark the possibilities for further research, I have inserted many "Questions", as well as "Exercises" that lead to illuminating examples. Theorems for which I do not give proofs are sometimes titled "Exerceorem", to stress their role in the development of the subject. I have also inserted some "Projects", which are designed as exercises to guide the reader in beginning their own research on the topic. I have not, however, lost any sleep over whether to call each result a theorem, proposition, exerceorem, or something else, so readers should neither deduce too much from the title given to a result nor be unduly disturbed by any inconsistencies they may discover – this beautiful English sentence I have adopted from the introduction of John Howie's *An Introduction to Semigroup Theory*, published by Academic Press in 1976.

Homomorphisms, especially endomorphisms, form a common thread throughout the book; you will meet this concept in almost all the chapters. Another focal point is the standard part of algebraic graph theory dealing with matrices and eigenvalues. In some parts of the book the presentation will be rather formal; my experience is that this can be very helpful to students in a field where concepts are often presented in an informal verbal manner and with varying terminology.

Content of the chapters

We begin, in Chapter 1, with basic definitions, concepts and results. This chapter is very important, as standard terminology is far from being established in graph theory. One reason for this is that graph models are so extremely useful in a great number of applications in diverse fields. Many of the modelers are not mathematicians and have developed their own terminology and results, without necessarily caring much about existing theory. Chapter 1 contains some new variants of results on graph homomorphisms and the relations among them, connecting them, in turn, to the combinatorial structure of the graph.

Chapter 2 makes connections to linear algebra by discussing the different matrices associated to graphs. We then proceed to the characteristic polynomial and eigenvalues, topics that will be encountered again in Chapters 5 and 8. There is no intention to be complete, and the content of this chapter is presented at a relatively elementary level.

In Chapter 3 we introduce some basic concepts from category theory, focusing on what will be helpful for a better understanding of graph concepts.

In Chapter 4 we look at graphs and their homomorphisms, in particular binary operations such as unions, amalgams, products and tensor products; for the latter two operations I use the illustrative names cross product and box product. It turns out that, except for the lexicographic products and the corona, all of these operations have a category-theoretical meaning. Moreover, adjointness leads to so-called Mor constructions; some of the ones presented in this chapter are new, as far as I know, and I call them diamond and power products.

In Chapter 5 we focus on unary operations such as the total graph, the tree graph and, principally, line graphs. Line graphs are dealt with in some detail; in particular, their spectra are discussed. Possible functorial properties are left for further investigation.

In Chapter 6, the fruitful notion of duality, known from and used in linear algebra, is illustrated with the so-called cycle and cocycle spaces. We then apply the concepts to derive Kirchhoff's laws and to "square the rectangle". The chapter finishes with a short survey of applications to transportation networks.

Chapter 7 discusses several connections between graphs and groups and, more generally, semigroups or monoids. We start with Cayley graphs and Frucht-type results, which are also generalized to monoids. We give results relating the groups to combinatorial properties of the graph as well as to algebraic aspects of the graph.

In Chapter 8 we continue the investigation of eigenvalues and the characteristic polynomial begun in Chapters 2 and 5. Here we present more of the standard results. Many of the proofs in this chapter are omitted, and sometimes we mention only the idea of the proof.

In Chapter 9 we present some results on endomorphism monoids of graphs. We study von Neumann regularity of endomorphisms of bipartite graphs, locally strong endomorphisms of paths, and strong monoids of arbitrary graphs. The chapter includes a fairly complete analysis of the strong monoid, with the help of lexicographic products on the graph side and wreath products on the monoid side.

In Chapter 10 we discuss unretractivities, i.e. under what conditions on the graph do its different endomorphism sets coincide? We also investigate questions such as how the monoids of composed graphs (e.g. product graphs) relate to algebraic compositions (e.g. products) of the monoids of the components. This type of question can be interpreted as follows: when is the formation of the monoid product-preserving?

In Chapter 11 we come back to the formation of Cayley graphs of a group or semigroup. This procedure can be considered as a functor. As a side line, we investigate (in Section 11.2) preservation and reflection properties of the Cayley functor. This is applied to Cayley graphs of right and left groups and is used to characterize Cayley graphs of certain completely regular semigroups and strong semilattices of semigroups.

In Chapter 12 we resume the investigation of transitivity questions from Chapter 8 for Cayley graphs of strong semilattices of semigroups, which may be groups or right or left groups. We start with Aut- and ColAut-vertex transitivities and finish with endomorphism vertex transitivity. Detailed examples are used to illustrate the results and open problems.

Chapter 13 considers a more topological question: what are planar semigroups? This concerns extending the notion of planarity from groups to semigroups. We choose semigroups that are close to groups, i.e. which are unions of groups with some additional properties. So we investigate right groups and Clifford semigroups, which were introduced in Chapter 9. We note that the more topological questions about planarity, embeddings on surfaces of higher genus or colorings are touched on only briefly in this book. We use some of the results in certain places where they relate to algebraic analysis of graphs – the main instances are planarity in Section 6.4 and Chapter 13, and the chromatic number in Chapter 7 and some other places.

Each chapter ends with a “Comments” section, which mentions open problems and some ideas for further investigation at various levels of difficulty. I hope they will stimulate the reader’s interest.

How to use this book

The text is meant to provide a solid foundation for courses on algebraic graph theory. It is highly self-contained, and includes a brief introduction to categories and functors and even some aspects of semigroup theory.

Different courses can be taught based on this book. Some examples are listed below. In each case, the prerequisites are some basic knowledge of linear algebra.

- Chapters 1 through 8 – a course covering mainly the matrix aspects of algebraic graph theory.
- Chapters 1, 3, 4, 7 and 9 through 13 – a course focusing on the semigroup and monoid aspects.
- A course skipping everything on categories, namely Chapter 3, the theorems in Sections 4.1, 4.2, 4.3 and 4.6 (although the definitions and examples should be retained) and Sections 11.1 through 11.2.
- Complementary to the preceding option, it is also possible to use this text as a short and concrete introduction to categories and functors, with many (somewhat unusual) examples from graph theory, by selecting exactly those parts skipped above.

About the literature

The literature on graphs is enormous. In the bibliography at the end of the book, I give a list of reference books and monographs, almost all on graphs, ordered *chronologically* starting from 1936; it is by no means complete. As can be seen from the list, a growing number of books on graph theory are published each year. Works from this list are cited in the text by author name(s) and publication year enclosed in square brackets.

Here I list some books, not all on graphs, which are particularly relevant to this text; some of them are quite similar in content and are cited frequently.

- N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, Cambridge 1996.
- M. Behzad, G. Chartrand, L. Lesniak-Forster, *Graphs and Digraphs*, Prindle, Weber & Schmidt, Boston 1979. New (fifth) edition: G. Chartrand, L. Lesniak, P. Zhang, *Graphs and Digraphs*, Chapman and Hall, London 2010.
- D. Cvetković, M. Doob, H. Sachs, *Spectra of Graphs*, Academic Press, New York 1979.
- C. Godsil, G. Royle, *Algebraic Graph Theory*, Springer, New York 2001.
- G. Hahn, G. Sabidussi (eds.), *Graph Symmetry*, Kluwer, Dordrecht 1997.

- P. Hell, J. Nešetřil, *Graphs and Homomorphisms*, Oxford University Press, Oxford 2004.
- H. Herrlich, G. Strecker, *Category Theory*, Allyn and Bacon, Boston 1973.
- W. Imrich, S. Klavžar, *Product Graphs*, Wiley, New York 2000.
- R. Kaschek, U. Knauer (eds.), *Graph Asymmetries*, Discrete Mathematics 309 (special issue) (2009) 5349–5424.
- M. Kilp, U. Knauer, A. V. Mikhalev, *Monoids, Acts and Categories*, De Gruyter, Berlin 2000.
- M. Petrich, N. Reilly, *Completely Regular Semigroups*, Wiley, New York 1999.
- D. B. West, *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, NJ 2001.

Papers, theses, book chapters and other references are given in the text where they are used.

Acknowledgements

This book was originally planned as a joint work with Lothar Budach. But the untimely death of Lothar, before we could really get started, cut short our collaboration.

First of all I thank the De Gruyter publishing house and especially the Mathematics Editor Friederike Dittberner, as well as Anja Möbius, for their cooperation and help.

Thanks go to all mathematicians, living or dead, whose work I have cannibalized freely – starting with Horst Herrlich, from whose book *Axiom of Choice* (Springer, Heidelberg 2006) I have taken this sentence.

Moreover, I thank all the students who have contributed over the years to improving the course – especially Barbara Langfeld, who looked through a final version of the text. One of my first students was Roland Kaschek, who became a professor in New Zealand, and one of most recent was Xia Zhang, who is now a professor in China. I thank all the colleagues who have used parts of my notes for their own courses and lectures.

I thank Mati Kilp (Estonia) for his contributions to several parts of the book, and I am grateful to Kolja Knauer for his comments, suggestions and contributions specially to Chapter 13. I also thank Sayan Panma (Thailand) who contributed many results to Chapters 11 and 12 while he was a PhD student at Oldenburg.

I thank Theresia Meyer for doing much of the typesetting of earlier versions, drawing pictures and providing various other sorts of technical help.

Above all, I thank my wife Jane Richter for her many good ideas as a non-mathematician, her encouragement and her patience with me during the intense periods of work.

Oldenburg and Berlin, April 2011 Ulrich Knauer

Contents

Chapter 1
Directed and undirected graphs

In this chapter we collect some important basic concepts. These concepts are essential for all mathematical modeling based on graphs. The language and visual representations of graphs are such powerful tools that graph models can be encountered almost everywhere in mathematics and informatics, as well as in many other fields.

The most obvious phenomena that can be modeled by graphs are binary relations. Moreover, graphs and relations between objects in a formal sense can be considered the same. The concepts of graph theory also play a key role in the language of category theory, where we consider objects and morphisms.

It is not necessary to read this chapter first. A reader who is already familiar with the basic notions may just refer back to this chapter as needed for a review of the notation and concepts.

1.1 Formal description of graphs

We shall use the word "graph" to refer to both directed and undirected graphs. Only when discussing concepts or results that are specific to one of the two types of graph we will use the corresponding adjective explicitly. An edge of a graph will be denoted by (x, y); this notation will also be used for directed graphs, whereas an edge in the particular case of undirected graphs will be written as $\{x, y\}$.

Definition 1.1.1. A ***directed graph*** or ***digraph*** is a triple $G = (V, E, p)$ where V and E are sets and

$$p : E \to V^2$$

is a mapping. We call V the set of ***vertices*** or ***points*** and E the set of ***edges*** or ***arcs*** of the graph, and we will sometimes write these sets as $V(G)$ and $E(G)$. The mapping p is called the ***incidence mapping***.

The mapping p defines two more mappings $o, t : E \to V$ by $(o(e), t(e)) := p(e)$; these are also called incidence mappings. We call $o(e)$ the ***origin*** or ***source*** and $t(e)$ the ***tail*** or ***end*** of e.

As p defines the mappings o and t, these in turn define p by $p(e) := (o(e), t(e))$. We will mostly be using the first of the two alternatives

$$G = (V, E, p) \quad \text{or} \quad G = (V, E, o, t).$$

We say that the vertex v and the edge e are ***incident*** if v is the source or the tail of e. The edges e and e' are said to be ***incident*** if they have a common vertex.

An ***undirected graph*** is a triple $G = (V, E, p)$ such that

$$p : E \to \{\overline{V} \subseteq V \mid 1 \leq |\overline{V}| \leq 2\}.$$

An edge e with $o(e) = t(e)$ is called a ***loop***. A graph G is said to be ***loopless*** if it has no loops.

Let $G = (V, E, o, t)$ be a directed graph, let e be an edge, and let $u = o(e)$ and $v = t(e)$; then we also write $e : u \to v$. The vertices of graphs are drawn as points or circles; directed edges are arrows from one point to another, and undirected edges are lines, or sometimes two-sided arrows, joining two points. The name of the vertex or edge may be written in the circle or close to the point or edge.

Definition 1.1.2. Let $G = (V, E, p)$ be a graph. If p is injective, we call G a ***simple graph*** (or a ***graph without multiple edges***). If p is not injective, we call G a ***multigraph*** or ***multiple graph***; sometimes the term ***pseudograph*** is used.

If $G = (V, E, p)$ is a simple graph, we can consider E as a subset of V^2, identifying $p(E)$ with E. We then write $G = (V, E)$ or $G = (V_G, E_G)$, and for the edge e with $p(e) = (x, y)$ we write (x, y).

Simple graphs can now be defined as follows: a ***simple directed graph*** is a pair $G = (V, E)$ with $E \subseteq V^2 = V \times V$. Then we again call V the set of ***vertices*** and E the set of ***edges***.

A ***simple undirected graph*** is a simple directed graph $G = (V, E)$ such that

$$(x, y) \in E \Leftrightarrow (y, x) \in E.$$

The edge (x, y) may also be written as $\{x, y\}$ or xy.

Mappings $w : E \to W$ or $w : V \to W$ are called ***weight functions***. Here W is any set, called the ***set of weights***, and $w(x)$ is called the ***weight*** of the edge x or of the vertex x.

Definition 1.1.3. A ***path*** a from x to y or an $\boldsymbol{x, y}$ ***path*** in a graph G is a sequence $a = (e_1, e_2, \ldots, e_n)$ of edges with $o(e_1) = x$, $t(e_n) = y$ and $t(e_{i-1}) = o(e_i)$ for $i = 2, \ldots, n$. We write $a : x \to y$ and call x the ***start (origin, source)*** and y the ***end (tail, sink)*** of the path a. The sequence $x_0, \ldots, x_n$ is called the ***trace*** of the path a. The set $\{x_0, \ldots, x_n\}$ of all vertices of the trace is called the ***support of the path a***, denoted by $\operatorname{supp} a$.

A path is said to be ***simple*** if every vertex appears at most once on the path. A path is said to be ***closed***, or is called a ***cycle***, if the start and end of the path coincide. A simple closed path, i.e. a simple cycle, is called a ***circuit***. The words ***(simple) semipath, semicycle*** or ***semicircuit*** will be used if, in the sequence of edges, the tail or origin of each edge equals the origin or tail of the next edge. This means that at least two consecutive edges have opposite directions. The notions of trace and support remain unchanged. In a simple graph, every (semi)path is uniquely determined by its trace. We can describe a path also by its vertices $x_0, \ldots, x_n$ where (x_0, x_1), ..., (x_{n-1}, x_n) are edges of the path. For undirected graphs, the notions of path and semipath are identical.

For the sake of completeness we also mention the following definition: the ***trivial*** $\boldsymbol{x, x}$ ***path*** is the path consisting only of the vertex x. It is also called a ***lazy path***.

The reader should be aware that, in the literature, the words "cycle" and "circuit" are often used in different ways by different authors.

Lemma 1.1.4. *For $x, y \in G$, every x, y path contains a simple x, y path. Every cycle in G is the union of circuits.*

Proof. Take $x, y \in G$. Start on an x, y path from x and proceed until one vertex z is met for the second time. If this does not happen, we already have a simple path; otherwise, we have also traversed a circuit. Remove this circuit, together with all its vertices but z, from the path. Continuing this procedure yields a simple x, y path. If we start with a cycle, we remove one edge $e = (y, x)$, and this gives an x, y path. Now collect the circuits as before. At the end we have a simple x, y path, which together with e gives the last circuit. □

Definition 1.1.5. Let $G = (V, E)$, and let $a = (e_1, \ldots, e_r)$ be a path with $e_i \in E$. Then $\ell(a) := r$ is called the ***length*** of a.

We denote by $F(x, y)$ the set of all x, y paths in G. Then $d(x, y) := \min\{\ell(a) \mid a \in F(x, y)\}$ is called the ***distance*** from x to y.

We call $\operatorname{diam}(G) := \max_{x,y \in G} d(x, y)$ the ***diameter*** of G. The length of a shortest cycle of G is called the ***girth*** of G. In German the figurative word *Taillenweite*, meaning circumference of the waist, is used.

Remark 1.1.6. In connected, symmetric graphs the distance $d : V \times V \to \mathbb{R}_0^+$ is a metric, if we set $d(x, x) = 0$ for all $x \in V$. In this way, (V, d) becomes a metric space. If $\{\ell(a) \mid a \in F(x, y)\} = \emptyset$, then $d(x, y)$ is not defined. Often one sets $d(x, y) = \infty$ in this case.

Definition 1.1.7. For a vertex x of a graph G, the ***outset*** of x is the set

$$\operatorname{out}(x) := \operatorname{out}_G(x) := \{e \in E \mid o(e) = x\}.$$

The elements of

$$N^+(x) := N_G^+(x) := \{t(e) \mid e \in \operatorname{out}_G(x)\}$$

are called the ***successors*** of x in G. The ***outdegree*** of a vertex x is the number of successors of x; that is,

$$\overleftarrow{d}(x) = \operatorname{outdeg}(x) := |\operatorname{out}(x)|.$$

Definition 1.1.8. The graph $G^{\mathrm{op}} := (V, E, t, o)$ is called the ***opposite graph*** to G.

The ***inset*** of a vertex x is the outset of x in the opposite graph G^{op}, so

$$\mathrm{in}(x) = \mathrm{in}_G(x) := \mathrm{out}_{G^{\mathrm{op}}}(x) = \{e \in E \mid t(e) = x\}.$$

The elements of

$$N^-(x) := N_G^-(x) := N_{G^{\mathrm{op}}}^+(x) := \{o(e) \mid e \in \mathrm{in}_G(x)\}$$

are called ***predecessors*** of x in G. The ***indegree*** of a vertex x is the number of predecessors of x; that is,

$$\overrightarrow{d}(x) = \mathrm{indeg}(x) := |\mathrm{in}(x)|.$$

A vertex which is a successor or a predecessor of the vertex x is said to be ***adjacent to x***.

Definition 1.1.9. In an undirected graph G, a predecessor of a vertex x is at the same time a successor of x. Therefore, in this case, $\mathrm{in}(x) = \mathrm{out}(x)$ and $N(x) := N^+(x) = N^-(x)$. We call the elements of $N(x)$ the ***neighbors*** of x. Similarly, $\mathrm{indeg}(x) = \mathrm{outdeg}(x)$. The common value $d_G(x) = d(x) = \deg(x)$ is called the ***degree*** of x in G.

An undirected graph is said to be ***regular*** or d***-regular*** if all of its vertices have degree d.

1.2 Connectedness and equivalence relations

Here we make precise some very natural concepts, in particular, how to reach certain points from other points.

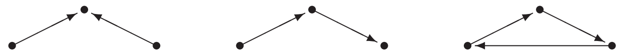
Definition 1.2.1. A directed graph G is said to be:

- ***weakly connected*** if for all $x, y \in V$ there exists a semipath from x to y;
- ***one-sided connected*** if for all $x, y \in V$ there exists a path from x to y or from y to x;
- ***strongly connected*** if for all $x, y \in V$ there exists a path from x to y and from y to x.

For undirected graphs, all of the above three concepts coincide. We then simply say that the graph is ***connected***; we shall also use this word as a common name for all three concepts.

If G satisfies none of the above three conditions, it is said to be ***unconnected*** or ***disconnected***.

Example 1.2.2. The following three graphs illustrate the three properties above, in the order given.



Definition 1.2.3. A connected graph is said to be ***n-vertex connected*** if at least n vertices must be removed to obtain an unconnected graph. Analogously, one can define ***n-edge connected*** graphs.

Remark 1.2.4. A ***binary relation*** on a set X is usually defined as a subset of the Cartesian product $X \times X$. This often bothers beginners, since it seems too simple a definition to cover all the complicated relations in the real world that one might wish to model. It is immediately clear, however, that every binary relation is a directed graph and vice versa. This is one reason that much of the literature on binary relations is actually about graphs. Arbitrary relations on a set can similarly be described by multigraphs.

An ***equivalence relation*** on a set X, i.e. a reflexive, symmetric and transitive binary relation in this setting, corresponds to a disjoint union of various graphs with loops at every vertex (reflexivity) which are undirected (symmetry), and such that any two vertices in each of the disjoint graphs are adjacent (transitivity). Note that the above-mentioned disjoint union is due to the fact that an equivalence relation on a set X provides a partition of the set X into disjoint subsets and vice versa.

1.3 Some special graphs

We now define some standard graphs. These come up everywhere, in virtually any discussion about graphs, so will serve as useful examples and counterexamples.

Definition 1.3.1. In the ***complete graph*** $K_n^{(l)}$ with n vertices and l loops, where $0 \leq l \leq n$, any two vertices are adjacent and l of the vertices have a loop.

The ***totally disconnected*** or ***discrete graph*** $\overline{K}_n^{(l)}$ with n vertices and l loops has no edges between distinct vertices and has loops at l vertices. If $l = 0$, we write K_n or $\overline{K}_n$.

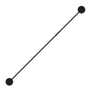
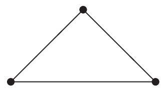
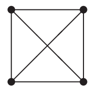
A simple, undirected path with n edges is denoted by P_n.

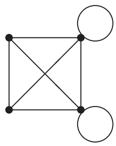
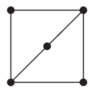
An undirected circuit with n edges is denoted by C_n.

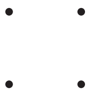
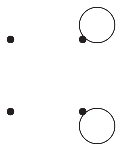
An ***r-partite graph*** admits a partition of the vertex set V into r disjoint subsets $V_1, \ldots, V_r$ such that no two vertices in one subset are adjacent.

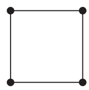
An r-partite graph is said to be ***complete r-partite*** if all pairs of vertices from different subsets are adjacent. The complete bipartite graph with $|V_1| = m$ and $|V_2| = n$ is denoted by $K_{m,n}$; similarly for complete r-partite graphs.

Example 1.3.2 (Some special graphs).

K_1: K_2:  K_3:  K_4: 

$K_4^{(2)}$:  $K_{2,3}$: 

$\overline{K}_4$:  $\overline{K}_4^{(2)}$: 

P_2: $C_3 = K_3$, $C_4 = K_{2,2}$: 

Definition 1.3.3. A graph without (semi)circuits is called a ***forest***. A connected forest is called a ***tree*** of G. A connected graph G' with the same vertex set as G is called a ***spanning tree*** if it is a tree. If G is not connected, the union of spanning trees for the components of G is called a ***spanning forest***.

Theorem 1.3.4. *Let G be a graph with n vertices. The following statements are equivalent:*

(i) *G is a tree.*

(ii) *G contains no semicircuits and has $n - 1$ edges.*

(iii) *G is weakly connected and has $n - 1$ edges.*

(iv) *Any two vertices of G are connected by a semipath.*

(v) *Adding any one edge produces exactly one semicircuit.*

Proof. We describe briefly the idea of the proof. Starting from some tree, i.e. statement (i), we verify (ii); then show the converse, that if (ii) does not hold then we cannot have a tree, and so on. □

Theorem 1.3.5. *A graph is bipartite if and only if it has no semicircuits with an odd number of edges.*

Proof. For "$\Rightarrow$", let $V = V_1 \bigcup V_2$. Since edges exist only between V_1 and V_2, all circuits must have an even number of edges.

For "$\Leftarrow$", let G be connected and take $x \in V$. Take V_1 to be the set of all vertices which can be reached from x along paths using an odd number of edges. Set $V_2 := V \setminus V_1$. If G is not connected, proceed in the same way with its connected parts. Isolated vertices can be assigned arbitrarily. $\square$

We recall the following definition: a pair $(P, \leq)$, where P is a set with a reflexive, antisymmetric, transitive binary relation $\leq$, is called a *partially ordered set* or a *poset*. We write $x < y$ if $x \leq y$ and $x \neq y$. We say that y *covers* x, written $x \prec y$, if $x < y$ and if $x \leq z < y$ implies $x = z$. See also Remark 1.2.4.

Proposition 1.3.6. *Every finite partially ordered set $(P, \leq)$ defines a simple directed graph H_P without cycles with vertex set P and edge set $\{(x, y) \mid x \prec y\}$, the so-called Hasse diagram of $(P, \leq)$, and conversely. Defining the edge set by $\{(y, x) \mid x \prec y\}$ gives a Hasse diagram where arcs are directed "down".*

Proof. A simple, directed graph H without cycles describes P completely, since $x \leq y$ if and only if either $x = y$ or there exists an x, y path in H whose edges (x_i, x_{i+1}) are interpreted as $x_i \prec x_{i+1}$.

For the converse we use analogous arguments. $\square$

Definition 1.3.7. A ***rooted tree*** is a triple $(T, \leq, r)$ such that:

- $(T, \leq)$ is a partially ordered set;
- H_T is a tree; and
- $r \in T$ is an element, the ***root of the tree***, where $x \leq r$ for all $x \in T$.

A ***marked rooted tree*** is a quadruple $(T, \leq, r, \lambda)$ such that $(T, \leq, r)$ is a rooted tree and $\lambda : T \to M$, with M being a set, is a mapping (weight function), which in this context is called the ***marking function***. We call $\lambda(x)$ a ***marking*** of x.

1.4 Homomorphisms

In mathematics, as in the real world, mappings produce images. In such images, certain aspects of the original may be suppressed, so that the image is in general simpler than the original. But some of the structures of the original, those which we want to study, should be preserved. Structure-preserving mappings are usually called homomorphisms. For graphs it turns out that preservation of different levels of structure or different intensities of preservation quite naturally lead to different types of homomorphism.

First, we give a very general definition of homomorphisms. We will then introduce the so-called covering, which has some importance in the field of informatics.

The general definition will then be specialized in various ways, and later we will use almost exclusively these variants. A reader who is not especially interested in the general aspects of homomorphisms may wish to start with Definition 1.4.3.

Definition 1.4.1. Let $G_1 = (V_1, E_1, o_1, t_1)$ and $G_2 = (V_2, E_2, o_2, t_2)$ be two directed graphs. A ***graph homomorphism*** $\theta : G_1 \to G_2$ is a pair $\theta = (\theta_V, \theta_E)$ of mappings

$$\theta_V : V_1 \to V_2$$
$$\theta_E : E_1 \to E_2$$

such that $o_2(\theta_E(e)) = \theta_V(o_1(e))$ and $t_2(\theta_E(e)) = \theta_V(t_1(e))$ for all $e \in E_1$.

If $\theta : G_1 \to G_2$ is a graph homomorphism and v is a vertex of G_1, then

$$\theta_E(\mathrm{out}_{G_1}(v)) \subseteq \mathrm{out}_{G_2}(\theta_V(v)) \quad \text{and} \quad \theta_E(\mathrm{in}_{G_1}(v)) \subseteq \mathrm{in}_{G_2}(\theta_V(v)).$$

Definition 1.4.2. If $\theta_E|_{\mathrm{out}_{G_1}(v)}$ is bijective for all $v \in V$, we call θ a ***covering*** of G_2. If $\theta_E|_{\mathrm{out}_{G_1}(v)}$ is only injective for all $v \in V$, then it is called a ***precovering***.

For simple directed or undirected graphs, we will mostly be working with the following formulations and concepts rather than the preceding two definitions.

The main idea is that homomorphisms have to preserve edges. If, in the following, we replace “homo” by “ega”, we have the possibility of identifying adjacent vertices as well. This could also be be achieved with usual homomorphisms if we consider graphs that have a loop at every vertex.

Definition 1.4.3. Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. A mapping $f : V \to V'$ is called a:

- ***graph homomorphism*** if $(x, y) \in E \Rightarrow (f(x), f(y)) \in E'$;
- ***graph egamorphism (weak homomorphism)*** if $(x, y) \in E$ and $f(x) \neq f(y) \Rightarrow (f(x), f(y)) \in E'$;
- ***graph comorphism (continuous graph mapping)*** if $(f(x), f(y)) \in E' \Rightarrow (x, y) \in E$;
- ***strong graph homomorphism*** if $(x, y) \in E \Leftrightarrow (f(x), f(y)) \in E'$;
- ***strong graph egamorphism*** if $(x, y) \in E$ and $f(x) \neq f(y) \Leftrightarrow (f(x), f(y)) \in E'$;
- ***graph isomorphism*** if f is a strong graph homomorphism and bijective or, equivalently, if f and f^{-1} are graph homomorphisms.

When $G = G'$, we use the prefixes “endo”, “auto” instead of “homo”, “iso” etc. We note that the term “continuous graph mapping” is borrowed from topology; there continuous mappings reflect open sets, whereas here they reflect edges.

Remark 1.4.4. Note that, in contrast to algebraic structures, bijective graph homomorphisms are not necessarily graph isomorphisms. This can be seen from Example 1.4.9; there the non-strong subgraph can be mapped bijectively onto the graph G without being isomorphic to it.

Remark 1.4.5. Note that for $f_0 \in \mathrm{EHom}(G, G')$, which identifies exactly two adjacent vertices, the graph $f_0(G)$ is also called an ***elementary contraction*** of G. The result of a series of elementary contractions $f_n(f_{n-1}(\dots(f_0(G))\dots))$ is usually called a ***contraction*** of G. This terminology is used mainly for the characterization of planar graphs (see Chapter 13).

Remark 1.4.6. Denote by $\mathrm{Hom}(G, G')$, $\mathrm{Com}(G, G')$, $\mathrm{EHom}(G, G')$, $\mathrm{SHom}(G, G')$, $\mathrm{SEHom}(G, G')$ and $\mathrm{Iso}(G, G')$ the homomorphism sets.

Analogously, let $\mathrm{End}(G)$, $\mathrm{EEnd}(G)$, $\mathrm{Cnd}(G)$, $\mathrm{SEnd}(G)$, $\mathrm{SEEnd}(G)$ and $\mathrm{Aut}(G)$ denote the respective sets when $G = G'$. These form monoids.

Indeed, $\mathrm{End}(G)$ and $\mathrm{SEnd}(G)$, as well as $\mathrm{EEnd}(G)$ and $\mathrm{SEEnd}(G)$, are monoids, i.e. sets with an associative multiplication (the composition of mappings) and an identity element (the identical mapping). Clearly, $\mathrm{End}(G)$ is closed. Also, $\mathrm{SEnd}(G)$ is closed, since for $f, g \in \mathrm{SEnd}(G)$ we get

$$(fg(x), fg(y)) \in E \overset{f \text{ strong}}{\Longleftrightarrow} (g(x), g(y)) \in E \overset{g \text{ strong}}{\Longleftrightarrow} (x, y) \in E.$$

The rest is clear.

Proposition 1.4.7. *Let G and G' be graphs and $f : G \to G'$ a graph isomorphism. For $x \in G$, we have* $\mathrm{indeg}(x) = \mathrm{indeg}(f(x))$ *and* $\mathrm{outdeg}(x) = \mathrm{outdeg}(f(x))$.

Proof. We prove the statement for undirected graphs.

As f is injective, we get $|N_G(x)| = |f(N_G(x))|$.

As f is a homomorphism, we get $f(N_G(x)) \subseteq N_{G'}(f(x))$, i.e. $|f(N_G(x))| \le |N_{G'}(f(x))|$.

As f is surjective, we have $N_{G'}(f(x)) \subseteq f(G)$; and, since f is strong, we get $|N_{G'}(f(x))| \le |N_G(x)|$.

Putting the above together, using the statements consecutively, we obtain $|N_G(x)| = |N_{G'}(f(x))|$.

Now we use $\deg(x) = |N_G(x)|$ and $\deg(f(x)) = |N_{G'}(f(x))|$ to get the result. $\square$

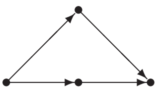
Subgraphs

The different sorts of homomorphisms lead to different sorts of subgraphs. First, let us explicitly define subgraphs and strong subgraphs.

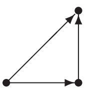
Definition 1.4.8. Let $G = (V, E)$. A graph $G' = (V', E')$ is called a ***subgraph*** (or *partial subgraph*) of G if there exists an injective graph homomorphism $f : V' \to V$.

A graph G' is called a ***strong subgraph*** (or *induced subgraph* or *vertex induced subgraph*) if there exists an injective strong graph homomorphism $f : V' \to V$.

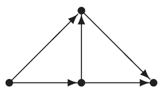
Example 1.4.9 (Subgraphs).



is a not strong subgraph while



is a strong subgraph of G:



Remark 1.4.10. A strong subgraph in general has fewer vertices than the original graph, but all edges of the original graph between these vertices are contained in the strong subgraph.

A subgraph in general contains fewer vertices and fewer edges than the original graph.

(Semi)paths, (semi)cycles and (semi)circuits are all subgraphs.

Definition 1.4.11. A ***strong***, ***one-sided*** or ***weak component*** of a graph is, respectively, a maximal strongly, one-sided or weakly connected subgraph.

A (strong) component is also called a ***clique*** of G. The number of vertices $\omega(G)$ of the largest clique of G is called the ***clique number*** of G.

See Example 1.2.2 for comparison.

The "edge dual" concept to a clique is a maximal independent subset of V.

Definition 1.4.12. Two vertices $x, y \in V$ are called ***independent vertices*** if $(x, y) \notin E$ and $(y, x) \notin E$. The ***vertex independence number*** is defined as

$$\beta_0(G) := \max\{|U| : U \subseteq V, \text{ independent}\}.$$

Analogously, two non-incident edges are called ***independent edges***, and we can define the ***edge independence number*** $\beta_1(G)$.

The elements of an independent edge set of G are also called 1-***factors*** of G; a maximal independent edge set of G is called a ***matching*** of G.

1.5 Half-, locally, quasi-strong and metric homomorphisms

In addition to the usual homomorphisms, we introduce the following four sorts of homomorphisms. As always, homomorphisms are used to investigate the structure of objects. The large number of different homomorphisms of graphs shows how rich and variable the structure of a graph can be. In Section 1.8 we summarize which of these homomorphisms have appeared where and under which names; we also suggest how they might be used in modeling.

The motivation for these other homomorphisms comes from the concept of strong homomorphisms or, more precisely, the notion of comorphism, i.e. the continuous mapping. A continuous mapping "reflects" edges of graphs. The following types of homomorphism reduce the intensity of reflection. In other words, an ordinary homomorphism $f : G \to G'$ does not reflect edges at all. This means it could happen that $(f(x), f(y))$ is an edge in G' even though (x, y) is not an edge in G, and there may not even exist any preimage of $f(x)$ which is adjacent to any preimage of $f(y)$ in G. The following three concepts "improve" this situation step by step.

From the definitions it will become clear that there exist intermediate steps that would refine the degree of reflection.

Definition 1.5.1. Let $G = (V, E)$ and $G' = (V', E')$ be graphs, and let $f \in \operatorname{Hom}(G, G')$. For $x, y \in V$, set

$$\begin{aligned} X &:= f^{-1}(f(x)),\\ Y &:= f^{-1}(f(y)). \end{aligned}$$

Let $(f(x), f(y)) \in E'$. Then f is said to be:

- ***half-strong*** if there exists $\tilde{x} \in X$ and $\tilde{y} \in Y$ such that $(\tilde{x}, \tilde{y}) \in E$;
- ***locally strong*** if $\begin{cases} \forall\, x \in X,\ \exists\, y_x \in Y \text{ such that } (x, y_x) \in E \text{ and} \\ \forall\, y \in Y,\ \exists\, x_y \in X \text{ such that } (x_y, y) \in E; \end{cases}$
- ***quasi-strong*** if $\begin{cases} \exists\, \tilde{x}_0 \in X \text{ such that } \forall\, \tilde{y} \in Y,\ (\tilde{x}_0, \tilde{y}) \in E \text{ and} \\ \exists\, \tilde{y}_0 \in Y \text{ such that } \forall\, \tilde{x} \in X,\ (\tilde{x}, \tilde{y}_0) \in E. \end{cases}$

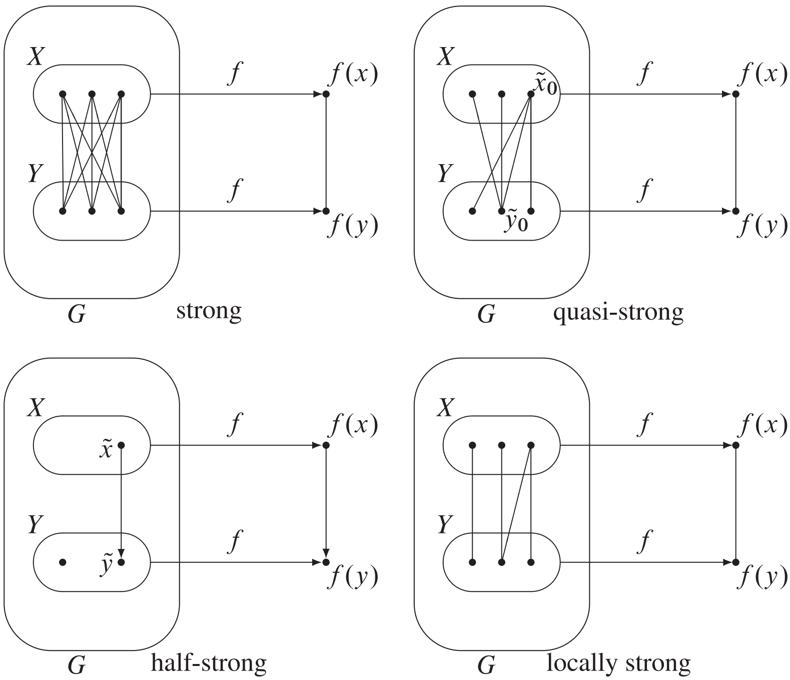
  We call $\tilde{x}_0$ and $\tilde{y}_0$ ***central vertices*** or, in the directed case, the ***central source*** and ***central sink*** in X and in Y with respect to $(f(x), f(y))$.

Remark 1.5.2. With the obvious notation, one has

$$\begin{aligned} \operatorname{Hom}(G, G') &\supseteq \operatorname{HHom}(G, G') \supseteq \operatorname{LHom}(G, G') \supseteq \operatorname{QHom}(G, G') \\ &\supseteq \operatorname{SHom}(G, G') \supseteq \operatorname{Iso}(G, G'), \\ \operatorname{End}(G) &\supseteq \operatorname{HEnd}(G) \supseteq \operatorname{LEnd}(G) \supseteq \operatorname{QEnd}(G) \\ &\supseteq \operatorname{SEnd}(G) \supseteq \operatorname{Aut}(G) \supseteq \{\mathrm{id}_G\}. \end{aligned}$$

Note that apart from $\mathrm{SEnd}(G)$, $\mathrm{Aut}(G)$ and $\{\mathrm{id}_G\}$, the other subsets of $\mathrm{End}(G)$ are, in general, not submonoids of $\mathrm{End}(G)$. We will talk about the ***group*** and the ***strong monoid*** of a graph, and about the ***quasi-strong monoid***, ***locally strong monoid*** and ***half-strong monoid*** of a graph if these really are monoids.

Example 1.5.3 (Different homomorphisms). We give three of the four examples for undirected graphs. The example for the half-strong homomorphism in the directed case shows that the other concepts can also be transferred to directed graphs.



From the definitions we immediately obtain the following theorem. To get an idea of the proof, one can refer to the graphs in Example 1.5.3.

Theorem 1.5.4. *Let $G \neq K_1$ be a bipartite graph with $V = V_1 \bigcup V_2$. Let (x_1, x_2) be an edge with $x_1 \in V_1$ and $x_2 \in V_2$. We define an endomorphism r of G by $r(V_1) = \{x_1\}$ and $r(V_2) = \{x_2\}$. Obviously, $r \in \mathrm{HEnd}(G)$. Moreover, the following hold:*

- $r \in \mathrm{LEnd}(G)$ *if and only if G has no isolated vertices;*
- $r \in \mathrm{QEnd}(G)$ *if and only if V_1 has a central vertex $\widetilde{x}_0$ with $N(\widetilde{x}_0) = V_2$ and correspondingly for V_2;*
- $r \in \mathrm{SEnd}(G)$ *if and only if G is complete bipartite.*

Proposition 1.5.5. *A non-injective endomorphism f of G is strong if and only if for all $x \in V$ with $f(x) = f(x')$ one has $N_G(x) = N_G(x')$.*

Note that for adjacent vertices x and x', this is possible only if both have loops.

Proof. Necessity is clear from the definition. Now suppose that $N_G(x) = N_G(x')$ for $x, x' \in V(G)$. Construct f by setting $f(x) = x'$ and $f(y) = y$ for all $y \neq x, x'$. It is clear that $f \in \mathrm{SEnd}(G)$. □

Corollary 1.5.6. *If* $\mathrm{Aut}(G) \neq \mathrm{SEnd}(G)$*, then* $|\mathrm{SEnd}(G) \setminus \mathrm{Aut}(G)|$ *contains at least two idempotents.*

Definition 1.5.7. A homomorphism f from G to G' is said to be ***metric*** if for any vertices $x, y \in V(G)$ there exist $x' \in f^{-1}f(x)$ and $y' \in f^{-1}f(y)$ such that $d(f(x), f(y)) = d(x', y')$. Denote by $\mathrm{MEnd}(G)$ the set of metric endomorphisms of G and by $\mathrm{Idpt}(G)$ the set of ***idempotent*** endomorphisms, i.e. $f \in \mathrm{End}(G)$ with $f^2 = f$, of G.

As usual we make the following definition.

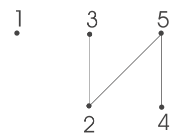
Definition 1.5.8. A homomorphism f from G to $f(G) \subseteq H$ is called a ***retraction*** if there exists an injective homomorphism g from $f(G)f$ to G such that $fg = \mathrm{id}_{f(G)}$. In this case $f(G)$ is called a ***retract*** of G, and then G is called a ***coretract*** of $f(G)$ while g is called a ***coretraction***.

If H is an unretractive retract of G, i.e. if $\mathrm{End}(H) = \mathrm{Aut}(H)$, then H is also called a ***core*** of G.

Remark 1.5.9. One has

$$\mathrm{Idpt}(G), \mathrm{LEnd}(G) \subseteq \mathrm{MEnd}(G) \subseteq \mathrm{HEnd}(G).$$

Example 1.5.10 (HEnd, LEnd, QEnd are not monoids). The sets HEnd, LEnd, QEnd are not closed with respect to composition of mappings. To see this, consider the following graph G



together with the mappings $f = \begin{pmatrix} 1&2&3&4&5 \\ 3&4&5&4&5 \end{pmatrix}$ and $g = \begin{pmatrix} 1&2&3&4&5 \\ 1&2&3&2&5 \end{pmatrix}$. Now $f \in \mathrm{QEnd}(G)$ and $g \in \mathrm{HEnd}(G)$ but $f^2 \in \mathrm{HEnd}(G) \setminus \mathrm{LEnd}(G)$ and $g \circ f \in \mathrm{End}(G) \setminus \mathrm{HEnd}(G)$. These properties are not changed if we add another vertex 0 to the graph which we make adjacent to every other vertex. The graph is then connected but no longer bipartite.

Question. Do Idpt and MEnd always form monoids? Can you describe graphs where this is the case?

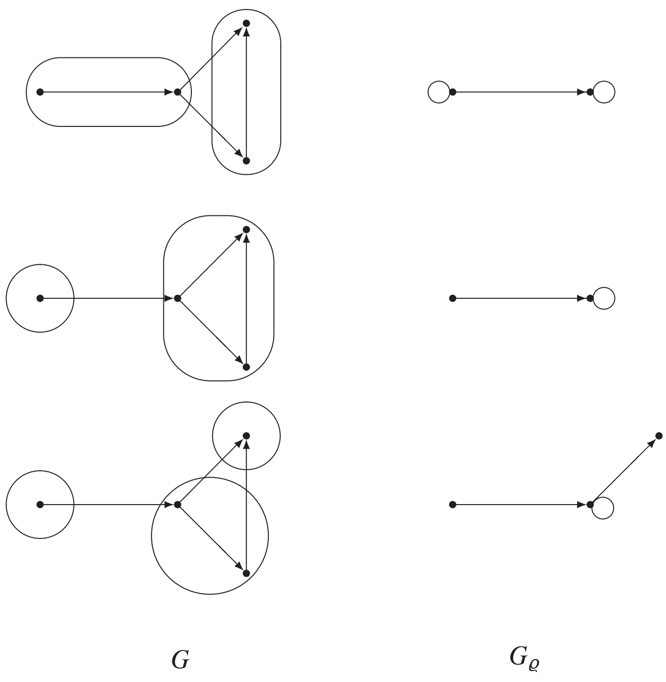
1.6 The factor graph, congruences, and the Homomorphism Theorem

The study of factor graphs by graph congruences turns out to be fundamental for the general investigation of homomorphisms. The connection to arbitrary homomorphisms is established through the canonical epimorphisms, and this leads to the Homomorphism Theorem for graphs. We formulate the theorem only for ordinary graph homomorphisms.

Factor graphs

Definition 1.6.1. Let $\varrho \subseteq V \times V$ be an equivalence relation on the vertex set V of a graph $G = (V, E)$, and denote by x_ϱ the equivalence class of $x \in E$ with respect to ϱ. Then $G_\varrho = (V_\varrho, E_\varrho)$ is called the ***factor graph*** of G with respect to ϱ, where $V_\varrho = V \big/ \varrho$ and $(x_\varrho, y_\varrho) \in E_\varrho$ if there exist $x' \in x_\varrho$ and $y' \in y_\varrho$ with $(x', y') \in E$, where ϱ is called a ***graph congruence***.

Example 1.6.2 (Congruence classes, factor graphs). We exhibit some graphs together with congruence classes (encircled vertices) and the corresponding factor graphs:



G $\qquad\qquad$ G_ϱ

Remark 1.6.3. By the definition of G_ϱ, the canonical epimorphism

$$\begin{aligned} \pi_\varrho \,:\, G &\to G_\varrho \\ x &\mapsto x_\varrho \end{aligned}$$

(which is always surjective) is a half-strong graph homomorphism.

Note that, in general, a graph congruence ϱ is just an equivalence relation. If we have a graph $G = (V, E)$ and a congruence $\varrho \subseteq V \times V$ such that there exist $x, y \in V$ with $(x, y) \in E$ and $x \,\varrho\, y$, then $(x_\varrho, x_\varrho) \in E_\varrho$, i.e. G_ϱ has loops.

If we want to use only loopless graphs, then $\pi_\varrho : G \to G_\varrho$ is a graph homomorphism only if

$$x \,\varrho\, y \Rightarrow (x, y) \notin E.$$

Therefore we make the following definition.

Definition 1.6.4. A ***(loop-free) graph congruence*** ϱ is an equivalence relation with the additional property that $x \,\varrho\, y \Rightarrow (x, y) \notin E$.

Definition 1.6.5. Let G_ϱ be the factor graph of G with respect to ϱ. If the canonical mapping $\pi_\varrho : G \to G_\varrho$ is a strong (respectively quasi-strong, locally strong or metric) graph homomorphism, then the graph congruence ϱ is called a ***strong (respectively quasi-strong, locally strong or metric) graph congruence***.

Example 1.6.6 (Connectedness relations). On $G = (V, E)$, with $x, y \in V$, consider the following relations:

$$\begin{aligned} x \,\varrho_1\, y &\Leftrightarrow \text{there exists an } x, y \text{ path and a } y, x \text{ path or } x = y; \\ x \,\varrho_2\, y &\Leftrightarrow \text{there exists an } x, y \text{ semipath or } x = y. \\ x \,\varrho_3\, y &\Leftrightarrow \text{there exists an } x, y \text{ path or a } y, x \text{ path.} \end{aligned}$$

The relation ϱ_1 is an equivalence relation; the factor graph G_{ϱ_1} is called a ***condensation*** of G.

The relation ϱ_2 is an equivalence relation; the factor graph G_{ϱ_2} consists only of isolated vertices with loops.

The relation ϱ_3 is not transitive and therefore not an equivalence relation.

The Homomorphism Theorem

For convenience we start with the so-called Mapping Theorem, i.e. the Homomorphism Theorem for sets, preceded by the usual result on mapping-induced congruence relations. Then, as for sets, we formulate the Homomorphism Theorem for graphs.

Proposition 1.6.7. *Let G and H be sets, and let $f : G \to H$ be a mapping. Using f we obtain an equivalence relation on G, the so-called* ***induced congruence****, if we define, for $x, y \in G$,*

$$x \,\varrho_f\, y \;\Leftrightarrow\; f(x) = f(y).$$

Moreover, by setting $\pi_{\varrho_f}(x) = x_{\varrho_f}$ for $x \in G$, we get a surjective mapping onto the factor set $G_{\varrho_f} = G/\varrho_f$. Here x_{ϱ_f} denotes the equivalence class of x with respect to ϱ_f and G/ϱ_f the set of all these equivalence classes.

Proof. It is straightforward to check that ϱ_f is reflexive, symmetric and transitive, i.e. it is an equivalence relation on G. Surjectivity of π_{ϱ_f} follows from the definition of the factor set. □

Proposition 1.6.8. *Let G and H be graphs, and let $f : G \to H$ be a graph homomorphism. Using f we obtain a graph congruence by defining, for $x, y \in V(G)$,*

$$x \,\varrho_f\, y \;\Leftrightarrow\; f(x) = f(y).$$

Moreover, by setting $\pi_{\varrho_f}(x) = x_{\varrho_f}$ for $x \in G$, we get a surjective graph homomorphism onto the factor graph $G_{\varrho_f} = G/\varrho_f$. Here x_{ϱ_f} denotes the congruence class of x with respect to ϱ_f and G_{ϱ_f} the factor graph formed by these congruence classes.

Proof. As for sets we know that ϱ_f is an equivalence relation and π_{ϱ_f} is a surjective mapping by Proposition 1.6.7. Now use Remark 1.6.3. □

Proposition 1.6.9 (The Homomorphism Theorem for sets). *For every mapping $f : G \to H$ from a set G to a set H, there exists exactly one injective mapping $\overline{f} : G_{\varrho_f} \to H$, with $\overline{f}(x_{\varrho_f}) = f(x)$ for $x \in G$, such that the following diagram is commutative, i.e. $f = \overline{f} \circ \pi_{\varrho_f}$:*

$$\begin{tikzcd}
G \arrow[r, "f"] \arrow[d, "\pi_{\varrho_f}"'] & H \\
G_{\varrho_f} \arrow[ur, "\overline{f}"'] &
\end{tikzcd}$$

Moreover, the following statements hold:

(a) *If f is surjective, then $\overline{f}$ is surjective.*

(b) *If we replace ϱ_f by an equivalence relation $\varrho \subseteq \varrho_f$, then $\overline{f} : G_\varrho \to H$ is defined in the same way, but is injective only if $\varrho = \varrho_f$.*