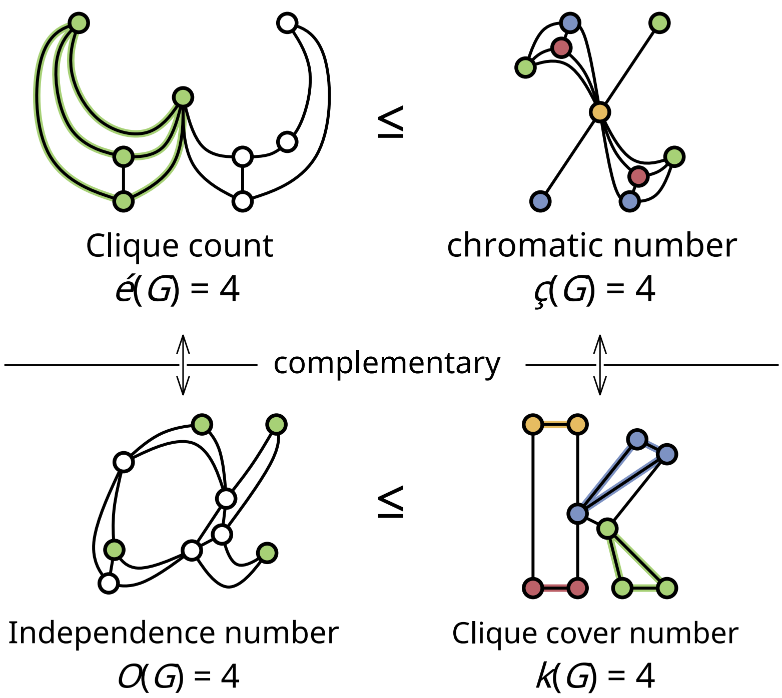
Algorithmic graph theory and Perfect graph

Notes on the lecture in the summer semester 2018

Prof. Dr. Ignaz Rutter

Summer semester 2018



Preface

This note was created for the first time parallel to the lecture “Algorithmic Graph Theory” in the winter semester 2014/15 at the Karlsruhe Institute of Technology (KIT). For the lecture in the summer semester 2018 at the University of Passau, the note will be revised and expanded to accompany the lecture. It is mainly based on the book “Algorithmic Graph Theory and Perfect Graphs” by Martin C. Golumbic, published by Elsevier in the series*Annals of Discrete Mathematics*. However, every now and then I have left out topics, streamlined them, and rephrased evidence. The lecture notes do not claim to be complete and should in particular*not*replace the lecture attendance. Even if I try very hard, I cannot guarantee that the content is correct. At this point I would like to thank all the lecture participants in the winter semester 2014/2015 at KIT and Thomas Bläsius, who held the associated exercise, read the notes very carefully, which meant that many errors could be eliminated, and who sent the " Cheat Sheet ”that adorns the front page.

Of course there are further corrections and suggestions for improvement, for example by email to rutter@fim.uni-passau.de ,always welcome.

Ignaz Rutter, March 13, 2018

table of contents

Chapter 1

introduction

In this chapter we first review some basic terms from graph theory and other basic areas. Following this, first examples of graph classes, graph presentations and algorithmic problems are presented, which are prototypical for the rest of the lecture.

Basically, this script assumes familiarity with basic concepts of complexity theory (especially O-Notation and NP-completeness) and graph theory. Nevertheless, individual terms are repeated for the sake of completeness and to standardize the notation.

1.1 Basic definitions and notation

amounts

Two sets A. and B. are *disjoint* if A. $\cap$ B = $\varnothing$ is applicable. In this case we write forC = A $\cup$ B. even C = A + B and mean that C = A $\cup$ B. and A. $\cap$ B = $\varnothing$ is applicable.

Relations

One *binary relation* on a lot X is an illustration R: X → P (X), whereby P (X) the power set of X designated. Can be equivalentR. than a lot $R \subseteq X \times X$ be understood by defining

$$(x, y) \in R \text{ exactly when } y \in R(x).$$

A relation can have one or more of the following properties:

Symmetry: $x \in R(y) \Rightarrow y \in R(x) \;\; \forall x, y \in X$, Asymmetry: $x \in R(y) \Rightarrow y \notin R(x) \quad \forall x, y \in X$,

Reflexivity: $x \in R(x) \quad \forall x \in X$,

Irreflectivity: $x \notin R(x) \quad \forall x \in X$,

Transitivity: $z \in R(y), y \in R(x) \Rightarrow z \in R(x) \quad \forall x, y, z \in X$.

Such a binary relation is a *Equivalence relation* when it is reflexive, symmetrical, and transitive. A binary relationR. is a strict partial order when it is irreflexive and transitive. (Show: Then isR. also asymmetrical.)

Graph

A graph G consists of one *Node set* V and an irreflexive binary relation V. The binary relation is represented by the figure Adj: $V \to P(V)$.For a knot $v \in V$ we denote by Adj (v) the amount of too v adjacent knot. The relation can be equivalent by a setE. $\subseteq V \times V$ represent. An orderly pair $(u, v) \in E$. called *Edge* from u after v. Obviously, $(u, v) \in E$.exactly when $v \in \mathrm{Adj}(u)$.

The present definition of graphs ensures that there is one graph for every two nodesu and v at most one edge (u, v) contains. In addition, it follows from the irreflectiveness that$(v, v) \notin E$ for all $v \in V$. The graphs appearing in the following are therefore always directed, but contain neither multiple edges nor so-called *Loops* the form(v, v).

For a node we define its *neighborhood* $N(v) = \{v\} + \mathrm{Adj}(v)$. In the following we shorten the notation for edges $(u, v) \in E$. to $uv \in E$. Two edges are called *adjacently*if they have a common endpoint.

May be G = (V, E) a graph with a set of nodes V and set of edges E. The graph $G^{-1} = (V, E^{-1})$ with

$$E^{-1} = \{(v, u) \mid (u, v) \in E\}$$

called *Inverse graph* from G. Obviously, $uv \in E$. exactly when $vu \in E^{-1}$. Of the *symmetrical conclusion* from G is the graph $\hat{G} = (V, \hat{E})$ with $\hat{E} = E \cup E^{-1}$. A graph is called*undirected* if its adjacency relation is symmetrical, that is, if $E^{-1} = E$is applicable.

A graph H = (V, F) is *oriented* if its adjacency relation is asymmetrical, that is, if $F \cap F^{-1} = \varnothing$. Also appliesF + F^{-1} = E, Is called H (or F) *orientation* from G.

May be G = (V, E) an undirected graph. That*complement* from G is the graph $\overline{G} = (V, \overline{E})$ with

$$\overline{E} = \{(u, v) \in V \times V \mid u \neq v \text{ and } (u, v) \notin E\}.$$

That is, two nodes are adjacent in G exactly when they are in $\overline{G}$ are not adjacent. A graph is *Completely* if every couple u, v of nodes with u *6th*= v is adjacent. The full graph withn Knot is made with K_n designated.

A graph H = (V′, E′) is a *Subgraph* of a graph G = (V, E) if both $V' \subseteq V$ as well as $E' \subseteq E$. Two types of subgraphs are of particular concern, namely those generated by a set of nodes or edges. May be $S \subseteq E$ a set of edges. The from S. *opened* Subgraph is the graph $H = (V_S, S)$ with $V_S = \{v \in V \mid \exists (u, v) \in S \vee \exists (v, u) \in S\}$. A set of knots $A \subseteq V$ *induced* a subgraph $G_A = (A, E_A)$ with

$$E_A = \{(u, v) \in E \mid u, v \in A\}.$$

Not every subgraph of a graph G is also an induced subgraph.

In the following we define a series of subgraphs and associated key figures that provide insight into the structure of a graph. The calculation of such structures and the associated key figures as well as their interaction will make up a considerable part of the lecture. It is therefore important to internalize these central concepts and the associated notation as well as possible. Be in the following G = (V, E) an undirected graph.

One *clique* is a set of nodes $A \subseteq V$, which induces a complete graph, that is $G_A \cong K_r$ for a $r \in \mathbb{N}$. It is $r = |A|$ the *size* the clique. A clique of size r also known as r-*Clique*. Obviously, a single node is always one 1-Clique. A clique is called *(inclusive) maximum* if she is not part of any larger clique. her name is *cardinality maximum* if G does not contain a clique with greater cardinality.

We denote the size of a maximum cardinality clique in G with ω (G); ω (G) called *Clique number* from G. One *Clique coverage* the size k is a partition of $V = A_1 + A_2 + \cdots + A_k$ so each A_i is a clique. We denote with k (G) the size of the smallest possible clique coverage of G; k (G) called *Clique Coverage Number* from G.

One *independent crowd* $X \subseteq V$ is a set of nodes that are not adjacent in pairs. We denote by α (G) the size of a cardinality-maximum independent set in G; α (G) called *Independence number* from G.

One *(real)* c-*Coloring* is a partition of the nodes $V = X_1 + X_2 + \cdots + X_c$ so each X_i is an independent set. We can then cut the nodes in X_i with the color i color to ensure that adjacent knots are different colors. We say that G c-is colorable. We denote by χ (G) the smallest c, so that G c-is colorable; χ (G) called *chromatic number* from G.

Obviously ω (G) ≤ χ (G) (Nodes of the same clique need different colors) and α (G) ≤ k (G) (Nodes of an independent set must be in different cliques of the overlap). In addition, $\omega(\overline{G}) = \alpha(G)$ and $\chi(\overline{G}) = k(G)$.

For any graph G = (V, E) let's define that *Starting* *Degree* $d_+(x) = |\mathrm{Adj}(x)|$ and the *Input degree* $d_-(x) = |\{y \in V \mid x \in \mathrm{Adj}(y)\}|$. It applies $\sum_{v \in V} d_+(v) =$

$\sum_{v \in V} d_-(v) = | E |$. Are called nodes with degree of entry 0 *source*, Node with output degree 0 are called *Sink*. Nodes with degree of entry and exit are called 0*isolated knot*. For undirected graphs we have$d_-(v) = d_+(v)$ for each knot $v \in V$. We simply refer to this number as *Degree* from v and write $d (v) = d_-(v) = d_+(v)$.

A *undirected path* the length l is a sequence of nodes $[v_0, \ldots, v_l]$ with $v_{i-1}v_i \in E$.or $v_iv_{i-1} \in E$. for i = 1,2,. . . , l.A *directed path* the length l is a sequence of nodes $[v_0, \ldots, v_l]$ with $v_{i-1}v_i \in E$. for i = 1,2,. . . , l.A path is called *simple*if it does not contain a node more than once.

A graph G is *coherent* if an undirected path exists between every two nodes. He is *strongly coherent* if between every two nodes x andy a directed path from x after y exists. One*Connected component* of an undirected graph is an inclusion-maximal connected subgraph.

Analog is a *(directed) circle* the length l + 1 a sequence of nodes $[v_0, \ldots, v_l, v_0]$ with$v_{i-1}v_i \in E$. for i = 1,. . . , land $v_lv_0 \in E$. A circle is called *simple*, if the v_i are different in pairs, ie, if v_i *6th*= v_j for i *6th*= j.

One *tendon* in a simple circle $[v_0, \ldots, v_l, v_0]$ is an edge $v_iv_j \in E$. in the i andj itself modulo `+ 1 differ by more than 1.

An undirected graph G = (V, E) is *bipartite* if its nodes can be partitioned into two disjoint independent sets, that is, $V = I_1 + I_2$ and every edge in E. has an endpoint in I_1 and an endpoint in I_2. We will write then$G = (I_1, I_2, E)$. Obvious is G bipartite if and only if $\chi (G) \leq 2$. A bipartite graph$G = (I_1, I_2, E)$ is *Completely* if for every two knots $x \in I_1$ and $y \in I_2$ the edge xy inE. is included.

The following graphs will come up again and again.

K_n: the full graph with n Node.C_n: the

circle of length n without tendons.P_n: the

path of length n without tendons.

$K_{n, m}$: the full bipartite graph with m + n Nodes partitioned into independent Sets of sizes n and m.

$K_{1, n}$ the *Star graph* with n + 1 Node.

mK_n: m disjoint copies of K_n.

1.2 Section graphs

May be *F.* a family of non-empty sets. Of the*Section graph* from *F.* is the graph with the set of nodes *F.* in which two knots $F, F' \in F$ are adjacent if and only if $F \cap F'$ *6th*= ∅.

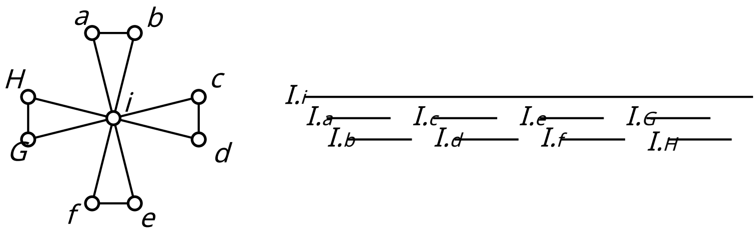


Figure 1.1: Windmill graph and associated interval representation.

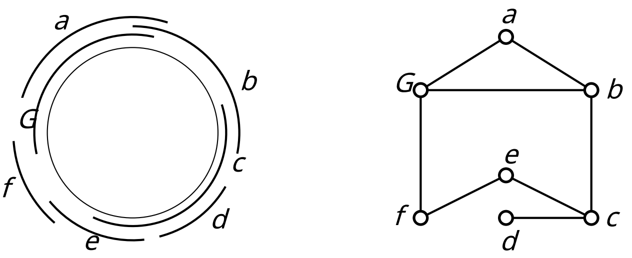


Figure 1.2: An arc graph.

Allowed *F.* To contain any set, any graph can be obtained as a slice graph of a suitable set system (exercise). In the following we therefore want to see the kind of sets that are included in*F.* may be included. If such a restriction has been specified, two natural questions arise from this, namely the*characterization* of those graphs that have such a sectional representation, and that *Recognition problem*to decide for a given graph whether it has a representation with the respective restriction.

In the following, some types of section representations are presented as examples. A slice graph of intervals of a linearly ordered set (sayR) called *Interval graph*; see figure 1.1. If all intervals have length 1, then it is a*Unit interval graph*. A*real interval graph* (engl. proper interval graph) is the intersection graph of a family of intervals with the property that no interval really contains another. One can show that unit interval graphs and real interval graphs form the same graph class (exercise).

A relaxation of this concept is obtained by looking at intervals on a finite segment and identifying the end points of the segment with one another so that they form a circle. The intervals then become arcs. We now also allow arcs that contain the connection point or go beyond it as sets and thus get the set of*Circular arc graph* (engl. circular-arc graph); see Figure 1.2. Analogous to the situation with interval graphs, a*real circular arc graph* the intersection graph of a family of arcs such that none of the arcs actually contains another.

A completely different generalization of interval graphs is obtained by first considering that interval graphs are exactly intersection graphs of partial paths of a path.

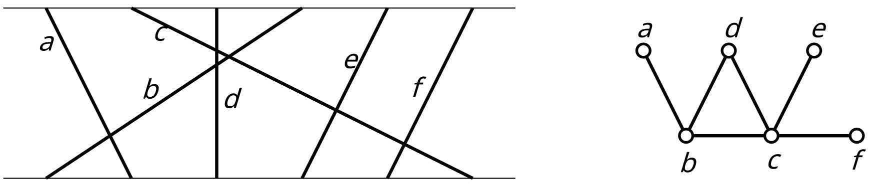


Figure 1.3: A permutation graph.

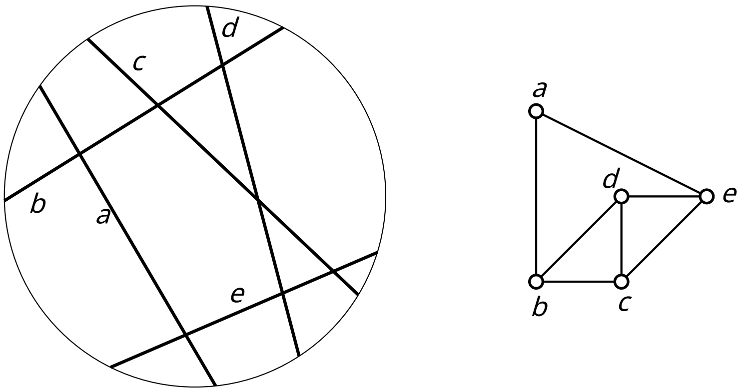


Figure 1.4: A circle chord graph.

that are. On the basis of this, the representation can be generalized by looking at section graphs of paths in trees or even of trees in trees (tree-in-tree).

Another completely different type of graph is created by considering two parallel straight lines and n Pick points on each of the straight lines as well n Lines that connect nodes on different straight lines with one another, so that an assignment (matching) is induced between the points. The section graph of the lines is called *Permutation graph*; see figure 1.3. We have, however2n Nodes distributed arbitrarily on a circle and a lot of n Lines that each form pairs of nodes, then these lines are chords of the circle and we get what are known as *Chord graph* (engl. circle graphs); see figure 1.4.

Comment 1.1. One can show that the class of true circular arc graphs (ie no interval contains another) is a true subset of circular chord graphs. That means that every real circular arc graph is also a circular chord graph, but there are also circular chord graphs that are not (real) circular arc graphs.

1.3 A short look at interval graphs

An undirected graph is a *Interval graph* if there is a bijective assignment of its nodes to a set *I.* of intervals in R. such that two nodes are and then are adjacent if the associated intervals have a non-empty cut. We then call*I.* one *Interval representation* from G.

An application. The following problem arises when assigning rooms to lectures. There are a lot of lectures, each with fixed times and a lot of rooms. The task now is to assign a room to each lecture so that at no time do two or more lectures take place in the same room.

This task can be modeled naturally as a graph problem. For this we consider the graphG = (V, E), which contains one node per lecture and in which two lectures are linked to one another if and when they overlap at any point in time. In order to obtain a valid room allocation, it is now sufficient to find a real coloring of this graph, whereby the colors then correspond to the individual rooms.

Usually such modeling is not very helpful because it is NP-difficult to decide whether a given number of colors is enough to find a true coloring. In this case, however, it is easy to see that the constructed graphG has further structure; G is an interval graph, as it was precisely defined as the intersection graph of the time intervals in which the lectures take place. As it turns out, the coloring problem can be solved efficiently on interval graphs.

Basic properties. Interval graphs have a number of basic properties which, in a similar form, also play a role for many other classes of intersection graphs.

Definition 1.2. A graph property is called *hereditary* if the fact that G possesses the property, implies that every induced subgraph of G owns the property.

For example, the property of being an interval graph is a *hereditary property*.

Proposition 1.3. *An induced subgraph of an interval graph is an interval graph.*

Proof. May be $\{I_v\}_{v \in V}$ an interval representation of G = (V, E). Then $\{I_v\}_{v \in X}$ an interval representation of the induced subgraph G_X. □

Note 1.4. The above proof in no way makes use of the fact that the sets of the intersection representations are intervals. In fact, all slice graphs are hereditary families.

Definition 1.5. A graph is called *chordal* when every circle longer than three has a chord.

Proposition 1.6. *Interval graphs are chordal.*

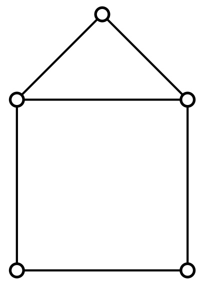


Figure 1.5: A non-chordal graph.

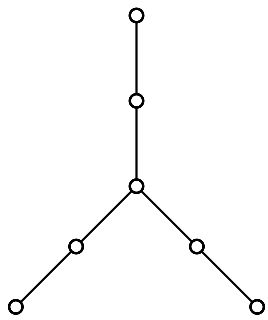


Figure 1.6: A chordal graph that is not an interval graph.

Proof. May be G an interval graph with a chordless circle $[v_0, v_1, \ldots, v_{l-1}, v_0]$ with $l > 3$.Denote I_k that the knot v_k corresponding interval. Choose $p_i \in I_{i-1} \cap I_i$ for $i = 1, \ldots, l-1$.There I_{i-1} and I_{i+1} do not intersect, they form p_i a strictly ascending or strictly descending sequence. It follows that I_0 and I_{l-1} don't cut yourself; a contradiction to the existence of the edge $v_0 v_{l-1}$. □

This gives us a first simple criterion to exclude certain graphs from being interval graphs; namely, if they are not chordal, such as the graph in Figure 1.5. On the other hand, there are graphs that are chordal but still have no interval representation, such as the graph in Figure 1.6. (Why?)

Definition 1.7. A graph is called *transitively orientable* if its edges can be directed in such a way that the resulting graph (V, F) meets the following conditions:

$$\text{away} \in F. \text{ and bc} \in F. \quad \text{implies} \quad ac \in F. \quad \forall ABC \in V$$

An undirected graph that is transitively orientable is also called *Comparability graph*. Figure 1.7 shows two examples.

Proposition 1.8. *The complement graph of an interval graph can be orientated transitively.*

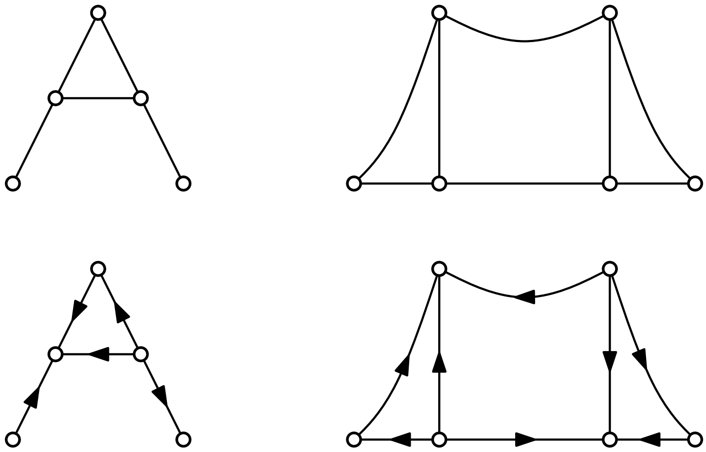


Figure 1.7: Comparability graph with transitive orientations.

Proof. May be $\{I_v\}_{v \in V}$ an interval representation of $G = (V, E)$. We define an orientation of the complement $\overline{G} = (V, \overline{E})$ as follows:

$$xy \in F \Leftrightarrow I_x < I_y \quad \forall xy \in \overline{E}.$$

Thereby means $I_x < I_y$ that the interval I_x completely to the left of the interval I_y lies. Obvious is the orientation F transitive, there $I_x < I_y < I_z$ implies that $I_x < I_z$. So is F a transitive orientation of $\overline{G}$. □

Similar to the chordal graphs, however, there are again graphs whose complements are comparability graphs, but which are not interval graphs. This condition is also necessary but not sufficient. As we shall see later, the conditions are sufficient together.

Theorem 1.9 (Gilmore, Hoffman 1964). *An undirected graph* G *is an interval graph if and only if it is chordal and its complement* $\overline{G}$ *is a comparability graph.*

If you look at the example graphs in Figures 1.1, 1.3, 1.4, 1.5 and 1.7, you can see that they can be colored with three colors. On the other hand, each of them contains a 3 clique. That is, for these graphs the clique number agrees with the chromatic number. This motivates the definitions of the following properties.

Definition 1.10. A graph G is called χ-perfect if for every induced subgraph G_A from G holds $\chi(G_A) = \omega(G_A)$.

Definition 1.11. A graph G is called α-perfect if for every induced subgraph G_A from G holds $\alpha(G_A) = k(G_A)$.

Chapter 2

Perfect graph

Let's look again at the following graph parameters:ω (G), the *Clique count* from G:

the size of a largest complete subgraph of G.

χ (G), the *chromatic number* from G: the minimum number of colors with the G can be dyed real. (equivalent: the minimum number of independent sets that is required to encompass all nodes ofG to cover.)

α (G), the *Independence number* from G: the maximum number of nodes in an independent set in G.

k (G), the *Clique cover number* from G: the minimum number of complete subgraphs required to encompass all nodes of G to cover.

The intersection of a clique and an independent set contains at most one element. Therefore applies

$$\omega (G) \leq \chi (G)$$

and

$$\alpha (G) \leq k (G).$$

These inequalities are dual to one another, since $\alpha (G) = \omega (\overline{G})$ and $k (G) = \chi (\overline{G})$.

Be now G = (V, E) an undirected graph. In the following we will deal with graphs that meet the following properties

$$\omega (G_{A.}) = \chi (G_{A.}) \qquad \text{for all A.} \subseteq V \qquad \text{(P1)}$$

and

$$\alpha (G_{A.}) = k (G_{A.}) \qquad \text{for all A.} \subseteq V. \qquad \text{(P2)}$$

Such graphs are called *Perfect*. According to the above duality it is clear that a graph fulfills property (P1) if and only if its complement graph$\overline{G}$ Property (P2) fulfilled. In the following we want to prove a much stronger statement, namely that statements (P1) and (P2) are equivalent.

2.1 The "Perfect Graph Theorem"

In this section we show that statements (P1) and (P2) are equivalent. For the proof we consider the further property

$$\omega(G_{A.}) \cdot \alpha(G_{A.}) \geq |A| \qquad \text{for all } A. \subseteq V \qquad \text{(P3)}$$

and show that it is equivalent to both (P1) and (P2). This property is motivated by the fact that on the one hand it is symmetrical with regard to complement formation. On the other hand, the statement is not surprising, as it says that the maximum clique and the maximum independent set cannot be small compared to the size of the graph at the same time. If both numbers were small, the graph wouldG nevertheless (P1) or (P2) can be fulfilled G cover it with a few, small independent quantities or with a few small cliques and would thus be small itself.

The so-called node multiplication will play an important role, as it allows us to "inflate" certain parts of the graph without changing its basic properties.

Definition 2.1. May be $G = (V, E)$ a graph and v a knot of G. The graph $G \circ v$ is the graph that you get from G obtained by making a new knot v' that adds to all neighbors of v connected is.

Lemma 2.2. *May be* $G = (V, E)$ *a graph. For* x *6th*$= y \in V$ *is applicable* $(G \circ x) - y = (G - y) \circ x$*.*

Proof. A practice. □

Definition 2.3. Be general $x_1, \dots, x_n$ the nodes of G and $h = (h_1, H_2, \dots, H_n)$ a vector with $H_i \in N_0$. The graph$H = G \circ H$ is constructed by looking at each node x_i by an independent set of H_i node $x^1_i, \dots, x^{H_i}_i$ replaced and x^s_i with x^t_j connects exactly when x_i and x_j in G are adjacent. We say that one$H$ by *Node multiplication* the end G receives.

Note 2.4. The definition allows explicit $H_i = 0$. In this case contains H no copy of x_i. In particular, one can find any induced subgraph of$G$ by multiplying by an appropriate one 0 / 1-Vector preserved.

Lemma 2.5. *May be* G *a graph with nodes* $x_1, \dots, x_n$*. is*$H \in N^n_0$ *a vector with* $H_i = 0$ *and* H' *the vector that made* H *by omitting the* i*-th component arises, then applies*

$$G \circ h = (G - x_i) \circ H'.$$

is H *a vector with* $H_i > 0$ *and* $H' = h - e_i$ *(*e_i *is the* i*-th unit vector) then applies*

$$G \circ h = (G \circ x_i) \circ H'.$$

Proof. A practice. □

Lemma 2.6. *May be* H *a graph that can be made up by node multiplication* G *receives. Then the following statements apply:*

(i) If G *Property (P1) fulfilled, then fulfilled* H *also (P1).*

(ii) If G *Property (P2) fulfilled, then fulfilled* H *also (P2).*

Proof. The proof is done by induction on the number of nodes. Obviously the statement is true ifG has only one knot. We now consider a graphG and assume that statements (i) and (ii) for all graphs with fewer nodes than G are valid. May beH = G ∘ H. Is one of the coordinates of H Zero, roughly $H_i = 0$, so you get H by node multiplication G− x_i. ThereG Property (P1) [resp. Property (P2)] fulfilled does this tooG - x_i. Hence (i) and (ii) follow from the induction hypothesis.

So we can assume that holds for every coordinate $H_i \geq 1$. Since every multiplication can be divided into individual steps according to Lemma 2.5, it is sufficient to calculate the result forH = G∘x to show. May bex′which by multiplying by x added nodes. Since every real induced subgraph ofG∘x using node multiplication from a real induced subgraph of G can be obtained, the respective statement by induction applies. So it suffices to show that ω (G ∘ x) = χ (G. ∘ x) (Statement (i)) or thatk (G ∘ x) = α (G ∘ x) (Statement (ii)).

Accepted G fulfilled (P1). Therex and x′are not adjacent, we have ω (G∘x) = ω (G). Consider a coloring of G with ω (G) Colours. Dyex′with the color of x. We get a coloring of G ∘ x with ω (G ∘ x) Colours. So statement (i) holds forG ∘ x.

Accepted G fulfilled (P2). It has to be shown that α (G ∘ x) = k (G ∘ x). May be $\mathcal{K}$ a clique cover of G with $|\mathcal{K}|$ =k (G) = α (G) and be $K_x \in \mathcal{K}$ the clique with $x \in K_x$. We distinguish between two cases.

Case 1: x *is in a cardinality-maximum independent set* S. *from* G *contain*, ie| S | = α (G).Then S.∪{x} an independent set of G∘x, so α (G∘x) = α (G) +1.

On the other hand is $\mathcal{K} \cup \{\{x\}\}$ a clique cover of G ∘ x. So it applies

$$k(G \circ x) \leq k(G) + 1 = \alpha(G) + 1 = \alpha(G \circ x) \leq k(G \circ x).$$

Hence α (G ∘ x) = k (G ∘ x).

Case 2: No cardinality maximum independent set of G *contains* x*.* Then α (G∘ x) = α (G). For each cardinality maximum independent set S. in G and every cliqueK $\in \mathcal{K}$ applies |S. ∩ K | = 1 (A practice!). This applies in particular toK_x. Here butx ∉ S is even true that every cardinality-maximum independent set S. exactly one element with the setD = $K_x \setminus \{x\}$ has in common. So we have α (G_{V-D}) = α (G) - 1. That implies that

$$k(G_{V-D}) = \alpha(G_{V-D}) = \alpha(G) - 1 = \alpha(G \circ x) - 1.$$

If you take a clique cover from G_{V-D} of size α (G ∘ x) - 1 along with the additional clique D. ∪{x}, so you get a clique cover of G ∘ x. So it appliesk (G ∘ x) = α (G ∘ x).

□

Lemma 2.7. *May be* G *an undirected graph, for which every real induced subgraph satisfies the property (P2) and is* H *a graph that can be made up by node multiplication*G *receives. Fulfills*G *Property (P3), this also applies to* H*.*

Proof. We do a proof by contradiction. May beH for this purpose a graph with a minimal number of nodes, which is derived from node multiplication G arises, but does not fulfill property (P3). Then applies

$$\omega(H)\,\alpha(H) < |X|, \tag{2.1}$$

whereby X is the set of nodes of H denotes, but property (P3) holds for every proper subgraph of H.

As in the previous lemma, we can assume that every node has been multiplied by at least 1 and that one node u with H ≥ 2 was multiplied. Be$U = \{u_1, \ldots, u_H\}$ the nodes of H, the u correspond. Because of the minimality ofH FulfillsH_{X-u_1} Property (P3), so

$$\begin{aligned} |X| - 1 = |X - u_1| &\leq \omega(H_{X-u_1})\,\alpha(H_{X-u_1}) && \text{[according to (P3)]} \\ &\leq \omega(H)\,\alpha(H) \\ &\leq |X| - 1 && \text{[according to (2.1)]} \end{aligned}$$

So equality applies everywhere and we define

$$p = \omega(H_{X-u_1}) = \omega(H), \; q = \alpha(H_{X-u_1}) = \alpha(H).$$

Then applies $pq = |X| - 1$. There H_{X-U} by node multiplication G − u can be obtained, fulfilled H_{X-U} by Lemma 2.6 the property (P2). The graphH_{X-U} so can with q complete subgraph $K_1, \ldots, K_q$ from H cover. Without limitation, we can assume that theK_i are sorted in pairs disjoint and non-ascending according to their size. It applies

$$\sum_{i=1}^{q} |K_i| = |X - U| = |X| - h = pq - (h - 1).$$

There $|K_i| \leq p$ can at most h − 1 the K_i less than p contribute to the total. So it applies

$$|K_1| = |K_2| = \cdots = |K_{q-h+1}| = p.$$

May be H' that of $X' = K_1 \cup \cdots \cup K_{q-h+1} \cup \{u_1\}$ induced subgraph of H. Then applies

$$|X'| = p(q - h + 1) + 1 < pq + 1 = |X|.$$

Due to the minimality of H follows that

$$\omega(H')\,\alpha(H') \geq |X'|.$$

Also applies p = ω (H) ≥ ω (H′), so

$$\alpha (H') \geq |X'| / p > q - h + 1.$$

Be now S.′an independent amount of size q − h + 2 in H′. It appliesu1 ∈S.′; otherwise would contain S.′two knots of a clique. But then it isS = S′∪U having an independent set q + 1 Knot in H. That is a contradiction to the definition of q. □

With this all important preliminary work is done and we come to the proof of the "Perfect Graph Theorem".

Theorem 2.8. *For an undirected graph* G = (V, E) *the following statements are equivalent:*

$$\omega (G_A) = \chi (G_A) \quad \text{for all } A \subseteq V \qquad (P1)$$
$$\alpha (G_A) = k (G_A) \quad \text{for all } A \subseteq V \qquad (P2)$$
$$\omega (G_A) \cdot \alpha (G_A) \geq |A| \quad \text{for all } A \subseteq V \qquad (P3)$$

Proof. For the proof we can assume that the theorem holds for all graphs that have fewer nodes than G.

(P1)⇒ (P3): Accepted, G_A can be expressed with ω (G_A) Coloring colors. Since it is at most α (G_A) Nodes of each color, it follows that ω (G_A) α (G_A) ≥ |A |.

(P3)⇒ (P1): Accepted G = (V, E) fulfills property (P3). Fulfilled by induction any real subgraph of G Properties (P1) - (P3). So it suffices to show that ω (G) = χ (G).

Suppose there was in G an independent set S. with ω (G_{V-S}) <ω (G). Then we could G_{V-S} with ω (G) −1 Color and color S. with an additional color and thus obtained a coloring with ω (G) Colors, so ω (G) = χ (G).

So we can assume that for any independent set S. the graph G_{V-S}an ω (G)-clique K (S) contains. May be*S.* the set of all independent sets of Gand note that S. ∩ K (S) = ∅. For eachx_i ∈V may be H_i the number of cliquesK (S), the x_i contain. May beH = (X, F) the graph that can be obtained by multiplying G obtained by x_i with H_i multiplied. From Lemma 2.7 it follows that

$$\omega (H)\, \alpha (H) \geq |X|.$$

On the other hand,

$$|X| = \sum_{x_i \in V} H_i \tag{2.2}$$

$$= \sum_{S. \in \mathcal{S}} |K(S)| = \omega(G)\,|\mathcal{S}.|, \tag{2.3}$$

$$\omega(H) \leq \omega(G), \tag{2.4}$$

$$\alpha(H) = \operatorname{Max}_{T \in \mathcal{S}} \sum_{x_i \in T} H_i \tag{2.5}$$

$$= \operatorname{Max}_{T \in \mathcal{S}} \sum_{S. \in \mathcal{S}} |T \cap K(S)| \tag{2.6}$$

$$\leq |\mathcal{S}.| - 1 \tag{2.7}$$

All in all, it follows that

$$\omega(H)\,\alpha(H) \leq \omega(G)\,(|\mathcal{S}.| - 1) < |X|,$$

a contradiction.

(P2)⇔ (P3): From the already proven implications it follows

$$G \text{ fulfilled (P2)} \Leftrightarrow \overline{G} \text{ fulfilled (P1)} \tag{2.8}$$

$$\Leftrightarrow \overline{G} \text{ fulfilled (P3)} \Leftrightarrow G \text{ fulfilled (P3)}. \tag{2.9}$$

□

Corollary 2.9. *A graph* G *is perfect if and only if its complement* $\overline{G}$ *is perfect.*

Corollary 2.10. *A graph* G *is perfect if and only if every graph* H *obtained by multiplying nodes* G *can get is perfect.*

2.2 p-Critical and partitionable graphs

It would be desirable to have a characterization of perfect graphs using forbidden substructures (cf. Kuratowski's theorem for planar graphs). Since the property of being perfect is hereditary, a characterization using forbidden induced subgraphs suggests itself. For this one is interested in the smallest possible imperfect graph. That motivates the following definition.

Definition 2.11. An undirected graph G called p-*critical* if it is minimally imperfect, G so is not perfect, but any real induced subgraph of G is perfect.

In particular, applies to one p-critical graph G

$$\alpha (G - x) = k (G - x) \text{ and } \omega (G - x) = \chi (G - x)$$

for each knot x. In the following we want to examine the structure of such graphs in more detail and thus formulate a conjecture about their exact structure. The corresponding assumption is called "Strong Perfect Graph Conjecture" and has since been proven (namely in 2006), it says that thep-critical graphs are exactly the circles of odd length and their complements. However, the entire proof comprises around 170 pages and is therefore not suitable for the lecture. Instead, we want at least some structure results forp-prove critical graphs and thus make the assumption plausible.

Theorem 2.12. *is* G *a* p*-critical graph, then*

$$n = \alpha (G) \omega (G) + 1,$$

and for all nodes x *from* G*,*

$$\alpha (G) = k (G - x) \textit{ and } \omega (G) = \chi (G - x).$$

Proof. According to Theorem 2.8, da G p-is critical n> α (G) ω (G) and n - 1 ≤ α (G - x) ω (G - x) for all nodes x. So

$$n - 1 \leq \alpha (G - x) \omega (G - x) \leq \alpha (G) \omega (G) < n.$$

It follows n − 1 = α (G) ω (G) as well as α (G) = α (G − x) = k (G − x) and ω (G) = ω (G − x) =χ (G - x). □

We now raise the properties found in Theorem 2.12 for definition.

Definition 2.13. Let α, ω ≥ 2 arbitrary integers. An undirected graphG withn Node is called (α, ω)-*partitionable* if n = αω + 1 and further for each node xfrom G is applicable

$$\alpha = k (G - x), \qquad \omega = \chi (G - x).$$

Theorem 2.12 shows that everyone p-critical graph G (α, ω) -is partitionable with α = α (G)and ω = ω (G). In fact, a more general statement applies.

Comment 2.14. May be G an (α, ω) -partitionable graph and x any node in G. Then owns G − x exactly αω node, has chromatic number ω and clique cover number α. An ω-coloring ofG - x thus partitions the nodes into ω independent sets, one of which must have a size at least α. Similarly, a minimal clique cover partitioned byG - x the nodes in α cliques, one of which must have at least size ω.

Theorem 2.15. *May be* G *a* (α, ω)*-partitionable graph. It applies*α = α (G) *and* ω = ω (G)*.*

Proof. May be $G = (V, E)$ (α, ω) -partitionable. According to Remark 2.14, α applies$\leq \alpha$ (G)and $\omega \leq \omega$ (G). For the converse, consider an independent set S. maximum size in G and be $y \in V - S$. Then S. a maximum independent set of $G - y$,so

$$\alpha (G) = | S | = \alpha (G - y) \leq k (G - y) = \alpha.$$

So $\alpha (G) \leq \alpha$.

Analogously, consider a maximal clique for ω K from G and $y \in V$ - K. Then K a maximum clique of G - y, so

$$\omega (G) = | K | = \omega (G - y) \leq \chi (G - y) = \omega.$$

It follows that $\omega (G) \leq \omega$. Overall, this results in $\alpha = \alpha (G)$ and $\omega = \omega (G)$. □

Theorem 2.15 shows that the numbers α and ω are uniquely determined for a partitionable graph. So in the following we only use the term*partitionable* and assume that $\alpha = \alpha (G)$ and $\omega = \omega (G)$.

Comment 2.16. The class of p-critical graph is a real subset of partitionable graphs, which in turn is a real subset of imperfect graphs.

Lemma 2.17. *is* G *a partitionable graph with* n *Node, then the following statements apply:*

(i) G *contains a lot of* n *maximum cliques* $K_1, K_2, \ldots, K_n$*that have every node of* G *precisely* ω (G) *times cover;*

(ii) G *contains a lot of* n *maximum independent quantities* $S._1, S._2, \ldots, S._n$*that have every node of* G *precisely* α (G) *times cover; and*

(iii) $K_i \cap S._j = \emptyset$ *exactly when* i = j.

Proof. Choose a maximum clique K from G and choose for each knot $x \in K$ a minimal clique cover $\mathcal{K}_x$ from G - x. According to Remark 2.14, all cliques are in $\mathcal{K}_x$Cliques of size ω. Be nowA. the $n \times n$-Matrix, the first line of which is the characteristic vector of the clique K is, and the other lines of which are the characteristic vectors of all cliques in $\mathcal{K}_x$ for each $x \in K$ are. (Note: these are ω different clique covers$\mathcal{K}_x$each consisting of α cliques. Together with the line forK so that is a total of $\omega\alpha + 1 = n$ Lines.)

Every knot $y \in$ / K is for everyone $x \in K$ exactly once from $\mathcal{K}_x$ covered. Every knotz $\in$ K is once by K covered and exactly once by each $\mathcal{K}_x$ with x *6th*= z. Each node is therefore covered exactly ω times. For each linea_i from A. denote K_i the corresponding clique with a characteristic vector a_i. We can write property (i) as1A =

ω1, whereby 1 denotes the line vector in which all entries 1 are. (Note: it still has to be shown that theK_i are different in pairs.)

We'll start with the construction of the $S._i$ away. Choose for each onei a knot $v \in K_i$ and denote with *S.* an optimal coloring (minimal coverage with independent sets!) of G - v. According to Remark 2.14, there is *S.* from ω disjoint independent sets of size α. Obviously, each of these sets contains at most one node ofK_i, and also has $K_i - v$ only ω−1 Knot so that a $S._i \in p$ exists with $K_i \cap S._i = \emptyset$. On the other hand isj *6th*= i, so applies $K_j \cap S._i$ *6th*= $\emptyset$ (so $|K_j \cap S._i| = 1$), otherwise*S.* would not be a valid coloring. This proves property (iii).

Now denote b_i the characteristic vector of $S._i$ and be B. the $n \times n$-Matrix whose rows have the b_i are (i = 1,. . . , n).According to the above observations, the following applies

$$a_i b_j^T = \begin{cases} 0 & i = j \\ 1 & \text{otherwise.} \end{cases}$$

We now consider the matrix AWAY$_T$. According to the above equation, the following applies AWAY$_T$ = J - I, wherebyI. is the identity matrix and J the matrix denotes whose entries are all 1 are. This matrix is not singular, so are theyA. and B. not, which in turn implies that the K_i and the $S._i$ must be different in pairs. Property (i) follows directly from this.

Furthermore applies

$$1B = 1BA_T(A._T)_{-1} = 1(J - I)(A_T)_{-1} = [(n - 1) / \omega]1 = \alpha 1,$$

what property (ii) proves. □

In fact, it can be shown that the cliques and independent sets constructed in Lemma 2.17 all maximal independent sets and all maximal cliques of G contain.

Lemma 2.18. *A partitionable graph* G *contains exactly* n *maximum cliques and* n *maximum independent sets.*

Proof. Be A. and B. the matrices whose rows form the characteristic vectors of the cliques and independent sets from Lemma 2.17. We assume thatc the characteristic vector of a maximal clique in G is. We'll show thatc a line of A. is.

First of all, $A(\omega_{-1}J - B_T) = \omega_{-1}AJ - AB_T = J - AB_T = I.$ and consequently

$$A._{-1} = \omega_{-1}J - B_T .$$

Let's consider a solution t the equation tA = c. It then applies

$$t = cA_{-1} = c(\omega_{-1}J - B_T) = \omega_{-1}cJ - cB_T = 1 - cB_T .$$

Hence is t a (0.1) -Vector. Also

applies

$$t1_T = (1 - cB_T)1_T = n - cB_T1_T = n - \alpha c1_T = n - \alpha\omega = 1.$$

So is t a unit vector and thus c a line of A. The statement for maximally independent sets follows analogously. □

Theorem 2.19. *May be* G *an undirected graph with* n *Knot and be* $\alpha = \alpha$ (G) *and*$\omega = \omega$ (G)*. The graph*G *is partitionable if and only if the following conditions apply:*

(i) $n = \alpha\omega + 1$*;*

(ii) G *has exactly* n *maximum cliques and* n *maximum independent sets;*

(iii) every node of G *is in exactly* ω *maximum cliques and in exactly* α *contain maximum independent quantities;*

(iv) each maximal clique cuts all but a maximal independent set and vice versa.

Proof. is G partitionable, the statements (i) - (iv) follow from the previous lemmas. So it still has to be shown that a graphG, which fulfills the properties (i) - (iv), is partitionable.

According to conditions (ii) - (iv) we can create the matrices A. and B. set up as before so that

$$AJ = JA = \omega J,\ BJ = JB = \alpha J,\ AB_T = J - I.$$

Be now x_i a knot of G and be H_T $_i$ the corresponding column in A. Because $A._TB = J - I$ follows that $H_iB = 1 - e_i$. This means,H_i chooses ω rows of B. from (these are independent sets!) that come together $G - x_i$ cover. So it's $\chi(G - x_i) \leq \omega$. Analogously, it can be shown that $k(G - x_i) \leq \alpha$ for all x_i. Here butn - 1 = αω must $\chi(G - x_i) = \omega$ andk $(G - x_i) = \alpha$ are valid. So isG partitionable. □

Corollary 2.20. *Everyone* p*-critical graph satisfies properties (i) - (iv) of Theorem 2.19.*

2.3 The strong “Perfect Graph Conjecture”

The odd circle $C._{2k+1}$ is for $k \geq 2$ not a perfect graph. Obviously, $\alpha(C._{2k+1})$ =k and k $(C_{2k+1}) = k + 1$ (or $\omega(C._{2k+1}) = 2$ and $\chi(C._{2k+1}) = 3$). On the other hand, every true subgraph is of $C._{2k+1}$ Perfect. So the odd circles (and therefore their complements) arep-critical.

The strong "Perfect Graph Conjecture" states that the above graphs are in fact the only ones p-critical graphs, i.e. a graph is perfect if and only if it does not contain any of the above graphs as an induced subgraph. The conjecture can be formulated in the following equivalent ways:

SPGC1 An undirected graph is perfect if and only if it is not an induced one Contains subgraph that goes to C_{2k+1} or $\overline{C}_{2k+1}$ with $k \geq 2$ is isomorphic.

SPGC2 An undirected graph G is perfect if and only if every odd circle the length at least 5 in G or $\overline{G}$ has a tendon.

SPGC3 The only ones p-critical graphs are C_{2k+1} and $\overline{C_{2k+1}}$ for $k \geq 2$.

In English the graphs are called C_{2k+1} and $\overline{C_{2k+1}}$ even *odd hole* and *odd anti-hole*. We already saw thatp-critical graphs have an extraordinarily high degree of symmetry. In particular, the previous section states that ifG a p-critical graph and $\alpha = \alpha(G)$ and $\omega = \omega(G)$, then the following statements must be fulfilled.

1. $n = \alpha\omega + 1$
2. Every node lies in exactly ω maximal cliques.
3. Every node lies in exactly α maximal independent sets.

4th G has exactly n maximum cliques (of size ω).

5. G has exactly n maximal independent sets (of size α).
6. The maximum cliques and maximum independent sets can be found withK_1 ,. . . , K_n and S_1,. . . , S_n number them so that $|K_i \cap S_j| = 1 - \delta_{ij}$, whereby$\delta_{ij}$ called the Kronecker Delta.

Obviously everyone is p-critical graph connected. It's easy to see thatC_n the only connected graph with n Is a node for which it holds that $\omega = 2$and that exactly n Has edges (maximum cliques) and for which each node is exactly two edges incident (each node is contained in exactly two maximum cliques). With this we can reformulate the conjecture again.

SPGC4th There is no p-critical graph with $\alpha > 2$ and $\omega > 2$.

Based on the families C_{2k+1} and $\overline{C_{2k+1}}$ another family of par- Construct definable graphs. The graphC^d_n has knots v_1,. . . , v_n and v_i and v_j are connected by an edge if and only if i and j at most d differentiate. The indices are modulon considered. You can easily see that

$C^{\omega-1}_{\alpha\omega+1}$ is an (α, ω) -partitionable graph. For ω =2 so we keep the family C_{2k+1}, for α = 2 the family $\overline{C_{2k+1}}$. However, these graphs are for α>2 and ω> 2not p-critical.

Theorem 2.21 (without proof.). *For* $\alpha \geq 3$ *and* $\omega \geq 3$ *are the partitionable graphs* $C^{\omega-1}_{\alpha\omega+1}$ *not* p*-critical.*

This allows the assumption to be rephrased in an equivalent manner.

SPGC5 is G p-critical with α (G) = α and ω (G) = ω, so contains G an induced
Subgraph that is isomorphic to $C^{\omega-1}_{d\omega+1}$.

In the meantime the above assumption could be shown. However, the evidence is extremely extensive.

Theorem 2.22 ("Strong perfect graph theorem ", 2006). *An undirected graph is perfect if and only if it does not contain an induced subgraph that leads to* C_{2k+1} *or* $\overline{C_{2k+1}}$ *with* $k \geq 2$ *is isomorphic.*

Chapter 3

Chordal graphs

One of the first graph classes for which it could be shown that all graphs contained therein are perfect were the chordal graphs. Indeed, the insight that chordal graphs satisfy both property (P1) and property (P2) provided the basis for the conjecture that the two properties are possibly equivalent. In this sense, studying chordal graphs forms the basis for studying perfect graphs.

Definition 3.1. A graph G is *chordal* if every circle of length greater than 3 has a chord.

This is equivalent to requiring that G does not have an induced subgraph that is isomorphic to C_n with $n > 3$. So it is obvious that being chordal is a hereditary trait. In Chapter 1 it was already mentioned that interval graphs are a special class of chordal graphs. Figure 3.1 shows two graphs of which the left is chordal, but the right is not.

3.1 Characterization

In the following we want to develop a characterization of chordal graphs, which will also be of central importance from an algorithmic point of view.

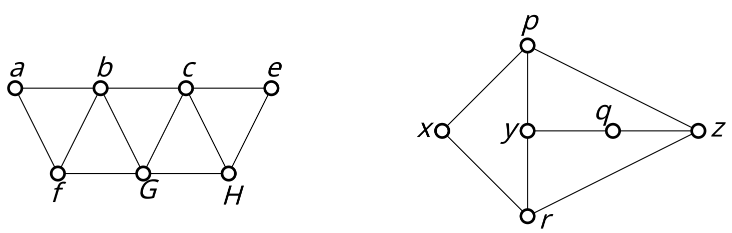


Figure 3.1: Two example graphs, the left one is chordal, the right one is not.

Definition 3.2. One knot x from G called *simplistic* if its adjacent nodes Adj (x) a complete subgraph of G form, that is, Adj (x) is a (not necessarily maximal) clique.

In the graph on the left in Figure 3.1 are the nodes a and e the only simplicial knots. The graph on the right has no simplicial node.

Definition 3.3. May be G = (V, E) an undirected graph and let σ = [v_1,. . . , v_n] a knot order. The order σ is a *perfect elimination scheme* if any knot v_i a simplicial node of the induced subgraph $G_{\{v_i, \ldots, v_n\}}$ is. Equivalent: any of the quantities $X_i = \{v_j \in Adj(v_i) \mid j > i\}$ is complete.

Obviously, the graph on the right in Figure 3.1 does not have a perfect elimination scheme because it does not have a simplicial node. For example, for the graph on the left,[a, f, b, g, c, h, e] a perfect elimination scheme. However, this is by no means unambiguous; in fact, the left graph 96 has different perfect elimination schemes.

A set of knots S. ⊆ V is a *(Knot) separator* for two non-adjacent knots aand b (even a-b-*Separator* called) if a and b in various related components of G - S lie.

Lemma 3.4. *is* G *a chordal graph, every (inclusion) minimal knot separator induces a complete subgraph of* G*.*

Proof. May be S. an (inclusion) minimum away-Separator from G with related components G_A. respectively. G_B. from G_{V-S}, the a respectively. b contain. Due to the minimality ofS.every node of S. a neighbor in A. and a neighbor in B. For every pair of knots x, y ∈ S. is there a pathx, a_1,. . . , a_r, y] and a path [y, b_1,. . . , b_t, x]with a_i ∈ A. and b_i ∈ B, so that these paths are of minimal length. Then the circle[x, a_1,. . . , a_r, y, b_1,. . . , b_t, x] simple and has a length of at least 4; so he must have a tendon. ThereS. but a separator is is a_ib_j ∈ / E. In addition, it follows from the minimality from r and t, that a_ia_j ∈/ E and b_ib_j ∈ / E for i <j - 1. Hence the only possible one tendon xy ∈ E. □

Note 3.5. It also follows that r = t = 1, which in turn implies that it is for every two nodes x, y ∈ S. Knot in A. and B. there that both too x as well as too yare adjacent.

In fact, it can even be shown that in A. and B. there is at least one node that is adjacent to all nodes of the separator. (A practice!)

Lemma 3.6. *Any chordal graph* G = (V, E) *has a simplicial knot. Is also* G *no clique, so owns* G *two nonadjacent simplicial knots.*

Proof. if G is complete, the statement trivially holds. So let's assume thatG two non-adjacent knots a and b and the statement holds for all graphs, which have fewer knots than G. May be S. an (inclusion) minimal knot separator for a and b and be G_A. respectively. G_B. the connected components of G_{V-S}, the a respectively. bcontain. By induction contains eitherG_{A+S} two nonadjacent simplicial nodes (then at least one of them is in A, there S. induces a complete set according to Lemma 3.4) or G_{A+S} is itself complete, and so every node is in A. simplicial inG_{A+S}. Since Adj (A) $\subseteq$ A + S, is a simplicial knot of G_{A+S} in A. also simplicial in G.The same argument shows that too B. contains a simplicial knot. □

Theorem 3.7. *May be* G *an undirected graph. The following statements are equivalent:*

(i) G *is chordal.*

(ii) G *has a perfect elimination scheme.*

(iii) Each (inclusion) minimal knot separator induces a complete subgraph of G*.*

Proof. (i) ⇒ (iii): is exactly the statement of Lemma 3.4.

(iii) ⇒ (i): May be [a, x, b, y_1, y_2,. . . , y_k, a] with k $\geq$ 1 a simple circle of G = (V, E).Every minimal away-Contains separator x and y_i for any i. But then it is$xy_i \in$E. a chord of the circle.

(i) ⇒ (ii): According to Lemma 3.6, has a chordal graph G a simplicial knot x. Since the graph $G_{V-\{x\}}$ is chordal and less than G, he has a perfect elimination scheme according to the induction hypothesis. The concatenation of [x] and this scheme then provides a perfect elimination scheme for G.

(ii) ⇒ (i): May be C. a simple circle of G and be x a knot of C. with minimal index in a perfect elimination scheme. There | Adj (x) $\cap$ C | $\geq$ 2 guarantees the simplicity of x a tendon in at the time of its removal C.

□

Note 3.8. Owns G a perfect elimination scheme and is v a simplicial knot of G, so there is also a perfect elimination scheme of G, that with v begins.

3.2 Recognition of chordal graphs

Theorem 3.7 and Remark 3.8 together result in an efficient algorithm for recognizing chordal graphs by iteratively searching for and removing a simplicial node. If all nodes can be removed from the graph in this way, the order of deletion results in a perfect elimination scheme. However, if the algorithm stops earlier, we have found an induced subgraph that does not contain a simplicial node. So the graph is not chordal. The main problem this

Input : Undirected graph $G = (V, E)$.
Output : Nodal order σ.
1 Assign the label to each knot $\varnothing$ to;2
for i ← n until 1 do
3 choose an unnumbered node v with the biggest label;σ (i) ←
4th v;
5 for every unnumbered node $w \in Adj(v)$ gap i to label (w) added;
6th end

Algorithm 1: LexBFS

The approach that goes back to Fulkerson and Gross lies in efficiency. Finding a simplicial node, for example, is easy inO $(n + m_2)$ Time to implement. If you then use the process forn Steps results in a total running time of O $(n_2 + nm_2)$. In the following we want to give a more efficient algorithm that solves the recognition problem for chordal graphs in linear time.

The basic intuition for this is as follows. The above procedure for constructing a perfect elimination scheme allows us to choose between at least two simplicial nodes at any point in time. So we can freely tie a knotv_n choose to keep it until the end. In a similar way, we can now also create another nodev_{n-1}who to v_n is adjacent, select and for the n − 1-Remove the th position. If we continued in this way, we would create a perfect elimination scheme in a "backwards" way. The algorithm presented below, which goes back to Rose, Tarjan and Lueker, follows exactly this idea.

The algorithm uses a so-called *lexicographic breadth-first search*. This is a breadth-first search that uses a special rule to resolve the ambiguities in choosing the next node to visit. The queue used to implement breadth-first search contains a set of unordered subsets (nodes whose order is arbitrary are in the same set). Occasionally these sets are refined, i.e. a set is broken down into several subsets and an order of the subsets is established, but two sets are never rearranged. The procedure is described in Algorithm 1. Each node has one*Label*which consists of a set of numbers listed in descending order. The nodes are then sorted lexicographically in the queue according to their labels.

Figure 3.3 shows an example of the sequence of a lexicographical breadth-first search on the graph from Figure 3.1. It turns out that the returned order [c, d, e, b, a] is a perfect elimination scheme. This is no coincidence and in fact always the case when the input graph is chordal.

For the proof denote $L_i(x)$ the label of x at the time of the selection of the i-th node (line 3 in the algorithm). The indexi is decreased in each pass. This means that the order of the nodes with different labels is always maintained. It

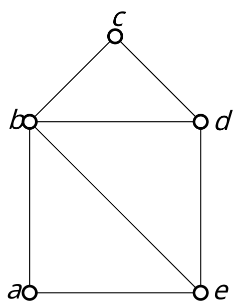


Figure 3.2: Chordal graph

	Label	number
a	∅	-
b	∅	-
c	∅	-
d	∅	-
e	∅	-

→

	Label	number
a	∅	5
b	{5}	-
c	∅	-
d	∅	-
e	{5}	-

→

	Label	number
a	∅	5
b	{5}	4th
c	{4th}	-
d	{4th}	-
e	{5, 4th}	-

→

	Label	number
a	∅	5
b	{5}	4th
c	{4th}	-
d	{4th, 3}	-
e	{5, 4th}	3

→

	Label	number
a	∅	5
b	{5}	4th
c	{4th, 2}	-
d	{4th, 3}	2
e	{5, 4th}	3

→

	Label	number
a	∅	5
b	{5}	4th
c	{4th, 2}	1
d	{4th, 3}	2
e	{5, 4th}	3

Figure 3.3: Chordal graph

the following properties apply:

(L1) $L._i(x) \leq L._j(x)$ $\forall j \leq i$(L2) $L._i(x) < L_i$

(y) $\Rightarrow$ $L._j(x) < L_j(y)$ $\forall j \leq i$

(L3) If σ applies$_{-1}$(a) <σ$_{-1}$(b) <σ$_{-1}$(c) and c $\in$ Adj (a) - Adj (b), so does one exist d $\in$ Adj (b) - Adj (a) with σ$_{-1}$(c) <σ$_{-1}$(d).

Property (L1) means that labels can only become larger, but not smaller. Property (L2) means that as soon as a label of one node is smaller than the label of another node, it remains that way in subsequent iterations. Property (L3) describes a condition under which a suitable noded must exist that before cwas numbered. The reasoning here is as follows: If there were no such knot, then there would have to be a and b at the time of numbering c have the same label. Because of the neighborhood ofa to c and the fact that b not to c is adjacent, this would lead to a a bigger label than b and, since the order of the labels does not change according to the other properties, so beforeb would have to be numbered. That is a contradiction.

We now show that lexicographic breadth-first search can be used to identify chordal graphs.

Theorem 3.9. *An undirected graph* G = (V, E) *is chordal if and only if the nodal order calculated by algorithm 1* σ *is a perfect elimination scheme.*

Proof. For $|V| = n = 1$the statement obviously applies. We assume that the statement for all graphs with less thann Node holds and consider an order σ obtained by applying algorithm 1 to a chordal graph G arises. It suffices to show by induction that$x = \sigma(1)$ a simplicial knot of G is.

So we assume that x is not simplicial. That is, there are nodes$x_1, x_2 \in \mathrm{Adj}(x)$ with $x_1 x_2 \notin E$. We choose these nodes so that x_2 maximum with respect to σ is. Now meet the knot$x = x_0, x_1, x_2$ just set the property (L3) so that there is a knot x_3 admit that x_1 is adjacent, but not too x_0. Again we choose$x_3$maximum with respect to σ with this property. Since, by assumption, the graph is chordal, so can the edgex_2x_3 does not exist. We now proceed inductively. We have a sequence$x_0, x_1, x_2, x_3, \ldots, x_m$ designed to meet the following characteristics.

(1) $x_0x_i \in E. \Leftrightarrow i \leq 2$,

(2) For $i, j > 0$ is applicable $x_ix_j \in E. \Leftrightarrow |i - j| = 2$,

(3) $\sigma^{-1}(x_0) < \sigma^{-1}(x_1) < \sigma^{-1}(x_2) < \cdots < \sigma^{-1}(x_m)$,

(4) For $j \geq 3$ is x_j is the maximum node with respect to σ $x_{j-2}x_j \in E$. but $x_{j-3}x_j \notin E$.

We just got the sequence for m = 3 constructed and now show how the construction can be continued. The knotsx_{m-2}, x_{m-1} and x_m satisfy property (L3), so there is a node x_{m+1} with $\sigma^{-1}(x_{m+1}) > \sigma^{-1}(x_m)$ that too x_{m-1} is adjacent, but not x_{m-2}. We votex $_{m+1}$ with this property such that it is maximal with respect to σ and show that the sequence $x_0, x_1, \ldots, x_{m+1}$ meets the above characteristics.

First is x_{m+1} by definition adjacent to x_{m-1} and not adjacent to x_{m-2}. Werex_{m+1} adjacent to x_{m-3}so we could use property (L3) x_{m-3}, x_{m-2} and x_{m+1} apply and get a knot larger than x_{m+1} and to x_{m-2} but not too x_{m-3} is adjacent. This would contradict the maximality ofx_m in (4). So it follows thatx_{m+1} not to x_{m-3} is adjacent. Werex_{m+1} to one of the nodes $x_0, \ldots, x_{m-4}$ orx_m neighboring, then according to (1) and (2) a circle without chords would result. But this would contradict the fact thatG is chordal.

The above inductive procedure continues indefinitely. But since the graph is finite, this leads to a contradiction. The knotx is so simplicial. From this it follows directly that a perfect elimination scheme is found for chordal graphs. The converse follows from Theorem 3.7. □

To get an efficient recognition algorithm for chordal graphs with linear running time, two more things are necessary. On the one hand it has to be shown that the lexicographical breadth-first search can be implemented with linear running time, on the other hand we need a procedure to decide in linear running time whether a given node order is a perfect elimination scheme.

3.2.1 Implementation of LexBFS

Instead of calculating the labels of the nodes, we simply keep the nodes in order according to their labels. That's why we define for a labell the amount S_l the not yet numbered node with label l. We use a queue Q, which the non-empty sets S_l in the correct order. Each of the quantities is saved as a doubly linked list. Contains at the beginningQ only one amount, namely $S_\emptyset = V$.For every knot v let's store a pointer SET (v), that points to the crowd that vcontains. It also savesv also a pointer to its position in the list of $S_{label(v)}$, so that v can be removed from the set in a constant time. The maps σ andσ^{-1} are represented by simple arrays. We also need a flag to control the processFLAG (p_l) for each of the quantities, which is initially set to 0, as well as a FIXLIST list of quantities $S_l$that require cleanup.

With such a data structure, it is easy to find a node with a maximum label. One chooses asv in line 3 of algorithm 1 any node in the last set in Q and removes this node from SET (v). If SET (v) hereby empty, we remove it from it Q. We now set σ and σ^{-1} for the knot v. Then we iterate over the neighbors of v and move them to the new resulting sets. One knotw $\in$Adj (v) with label l should be in the crowd $S_{l \cdot \sigma^{-1}(v)}$ moved into Q right after S_l follows. To avoid creating the same quantity multiple times, we use FLAG (SET (w)). We check whether the FLAG is set, and only if the flag is still 0, we create the set and set the flag to 1. After that, the target set is in any case Q directly behind SET (w). In order to be guaranteed to reset all flags at the end, we also add a pointer to SET (w)in FIXLIST. The actual changing of the knots is easy and constant. Then we iterate once over the sets of the FIXLIST, set their associated flags back to 0 and remove sets that have become emptyQ. Algorithm 2 shows the course of the procedure as pseudo-code. It's not hard to see the entire update justO (| Adj (v) |) Time needed. It follows that the total runtime of the detection algorithm inO (| V | + | E |) lies.

Theorem 3.10. *Algorithm 1 can be implemented to perform a lexicographic breadth first search on an undirected graph* G = (V, E) *with* O (| V | + | E |) *Time and space requirements calculated.*

3.2.2 Recognition of perfect elimination schemes

We can now determine a possible nodal order for a graph by means of lexicographical breadth-first search, which is a perfect elimination scheme if and only if the graph is chordal. So finally we still have to check whether the computed knot order is a perfect elimination scheme. So that this step does not become a bottleneck in the detection of chordal graphs, we need an algorithm that decides for a graph with a given nodal order whether

```
for all unnumbered nodes w ∈ Adj (v) do
    if FLAG (SET (w)) = 0 then
        Create new set S.′ and add them directly after SET (w) in Q a; FLAG
        (SET (w)) ← 1; FLAG (S.′) ← 0; Pointer to SET (w) in FIXLIST;
    end
    May be S.′ the amount immediately after SET (w) in
    Q;Remove w from SET (w); Gap w to S.′ added; SET(
    w) ← S.′
end
for a lot S. in FIXLIST do
    FLAG (S) ← 0;
    if S. empty then
        Remove S. the end Q;
    end
    Remove S. from FIXLIST;
end
```

Algorithm 2: Pseudocode of the update step from line 5 of algorithm 1.

it is a perfect elimination scheme. Of course, you can proceed naively and always check whether the following neighbors form a clique for each node. However, this requires at least a quadratic running time. On the other hand, it is quite obvious that this means that many neighborhoods are checked multiple times.

The basic idea behind the following algorithm is to go through the nodes in ascending order of numbering, check some neighborhoods and then remove them. To check the neighborhood of the first nodex forms a clique, we pass it on to the neighbor y from x with the smallest number the task of checking whether he is with all the neighbors of x connected is. This is only done when the knoty is processed. At this point the knot hasy may have received orders from other nodes to check neighborhoods.

Let us now consider the point in time when y. is y not to all neighbors of x adjacent, then the following neighbors of x no clique and the knot order is not a perfect elimination scheme. Otherwise the neighborhood is fromx without y in the remainder graph a subset of the neighborhood of y. is y simplicial, so this implies the simplicity of x. To treat x no further steps are necessary.

Algorithm 3 describes the procedure as pseudo code. The nodes treated are not removed explicitly, but rather implicitly through the exclusive consideration of nodes whose index in σ is greater than the current iteration.

Arrays are used around σ and σ^{-1} evaluate in constant time, for Adj (v)

```
1 Boolean procedure PERFECT (σ)2
Beginning
3     for all knots v do A (v) ← ∅;for i
4th   ← 1 until n - 1 do
5         v ← σ (i);
6th       X ← {x ∈Adj (v) | σ-1(v) <σ-1(x)};if X =
7th       ∅then go to line 10;u ← σ (min {σ-1
8th       (x) | x∈X});concatenate X - {u} to A
9         (u);if A (v) - Adj (v) 6th= ∅then
10
11            return not correct;
12th      end
13th  end
14th  return true;
15th end
```

Algorithm 3: Procedure for checking whether a given knot sequence σ is a perfect elimination scheme.

and A (v) doubly linked lists are used. The jump in line 7 will be exactj − 1 Times performed, whereby j denotes the number of connected components. The amountA (u) may contain duplicates. Algorithm 4 shows how the test on line 10 inO (| Adj (v) | + | A (v) |)Time can be done by using an array of size nis used whose entries are initial 0 are.

```
1 for w ∈Adj (v) do TEST(w) ← 1;2 for
w ∈A (v) do
3     if TEST(w) = 0 then
4th       return not empty;
5     end
6th end
7th for w ∈Adj (v) do TEST(w) ← 0;8th
return empty;
```

Algorithm 4: Adjacency test on line 10 of algorithm 3.

This means that the entire algorithm increases proportionally with runtime and memory

$$| V | + \sum_{v \in V} | \text{Adj} (v) | + \sum_{u \in V} | A (u) | .$$

Here denotes |A (u) | the size of the list A (u) at the time of processing u. It is not difficult to see that the middle term dominates the final term, and therefore the final two term in O (| E |) lie. The algorithm has a linear running time.

Theorem 3.11. *Algorithm 3 correctly checks whether the input order is correct* σ *is a perfect elimination scheme. The algorithm can be proportional to runtime and memory requirements* $|V| + |E|$ *implemented.*

Proof. The statements on complexity have already been proven. It remains to show the correctness of the procedure.

The algorithm delivers the statement “wrong” in the $\sigma_{-1}(u)$ -th iteration if and only if there are three nodes and many more gives ($\sigma_{-1}(v) < \sigma_{-1}(u) < \sigma_{-1}(w)$) whereby u in line 8 by the $\sigma_{-1}(v)$ -te iteration is defined, and u, w $\in$ Adj (v) but u is not adjacent to w.

Obviously, in the case of the answer “wrong”, there is no perfect elimination scheme. Conversely, let's assume that σ is not a perfect elimination scheme and the algorithm still returns “true”. May bev the node with maximum index $\sigma_{-1}(v)$,so that $X = \{w \mid w \in \text{Adj}(v) \text{ and } \sigma_{-1}(v) < \sigma_{-1}(w)\}$ is not complete; thus the last knot in the order that is not simplicial at the time of its removal. May beuthe knot of X, which in line 8 by the $\sigma_{-1}(v)$ -te iteration is defined. Then (in line 9)X - {u} to A (u) added. Since the algorithm does not abort, every node isx $\in$ X - {u} to u adjacent. Furthermore, due to the maximality of $\sigma_{-1}(v)$ the knot u simplicial, in particular two nodes are off X - {u} adjacent. So isX completely, which is contrary to the assumption.

□

Corollary 3.12. *The recognition problem for chordal graphs can be solved in linear time.*

3.3 Chordal graphs as section graphs

As we saw in Chapter 1, the interval graphs are a true subset of the chordal graphs. This naturally leads to the question of whether chordal graphs can be described as intersection graphs of a topological family that are somewhat more general than the intervals of a straight line. In this section we show that a graph is chordal if and only if it is the intersection graph of a family of subtrees of a tree; see Figure 3.4.

A family $\{T_i\}_{i \in I.}$ of subsets of a set T meets the so-called *Helly property*if the fact that for a subset J $\subseteq$ I. is applicable $T_i \cap T_j$ *6th*= $\varnothing$ for all i, j $\in$ J, implies that $\cap_{j \in J} T_j$ *6th*= $\varnothing$.

Proposition 3.13. *A family of subtrees of a tree fulfills the Helly property.*

Proof. We accept $T_i \cap T_j$ *6th*= $\varnothing$ for all i, j $\in$ J. We first consider the case that there are three nodes ABC from T so that for every two of the nodes there is a tree T_j withj $\in$ J exists that contains both. May beS. the set of indices s, so that T_s contains at least two of the three points, and are $P_{.1}$, P_2, P_3 the easy paths that a with b, b with

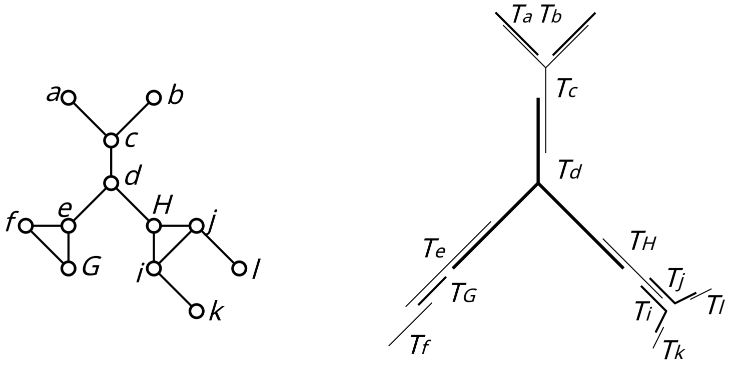


Figure 3.4: Chordal graph and a representation as a sectional graph of sub-trees of a tree.

c and c with a associate. ThereT is a tree, follows $P_{.1} \cap P_{.2} \cap P_{.3}$ *6th*= ∅. However, each containsT$_s$(see ∈S) at least one of the paths $P_{.i}$. So

$$\bigcap_{s \in S.} T_s \supseteq P_{.1} \cap P_{.2} \cap P_{.3} \; 6th\!= \varnothing.$$

We now prove the proposition by induction. We accept the statement

$$T_i \cap T_j \; 6th\!= \varnothing \text{ for all i, j} \in J \Rightarrow \bigcap_{j \in J} T_j \; 6th\!= \varnothing$$

applies to index sets J with the size at most k. The statement applies to k = 2.

Consider a family of subtrees {T_{i_1},. . . , $T_{i_{k+1}}$}. According to the induction hypothesis there is animal knot ABC from T, so that

$$a \in \bigcap_{j=1}^{k} T_{i_j}, \quad b \in \bigcap_{j=2}^{k+1} T_{i_j}, \; c \in T_{i_1} \cap T_{i_{k+1}} .$$

Each of the trees T_{ij} contains at least two of the nodes ABC. With the statement out follows the first paragraph of the proof $\bigcap_{j=1}^{k+1} T_{i_j} \; 6th\!= \varnothing$. □

Theorem 3.14. *May be* G = (V, E) *an undirected graph. The following statements are equivalent:*

(i) G *is chordal.*

(ii) G *is a section graph of a family of subtrees of a tree.*

(iii) There is a tree T = (*K,E.*)*, its node set K the maximum cliques of*G *are such that each of the induced subgraphs* T_{K_v} *(*v ∈V*) is connected (i.e. a subtree), where K*$_v$ *the amount of cliques in K is the*v *contain.*

Proof. (iii) ⇒ (ii) We assume there is a tree T = (*K*,*E*.), which has the properties from (iii) Fulfills. Bev, w ∈V. The knots v and w are in Gneighboring when there is a maximum clique A. ∈ *K* that contains both nodes. Again, this is the case if and only if $K_v \cap K_w$ *6th*= ∅, so exactly when $T_{K_v} \cap T_{K_w}$ *6th*= ∅. So isG Section graph of the family of subtrees $\{T_{K_v} \mid v \in V\}$.

(ii) ⇒ (i) May be $\{T_v\}_{v \in V}$ a family of subtrees of a tree T, so that vw ∈E. exactly when $T_v \cap T_w$ *6th*= ∅.

We assume that G a circle $[v_0, v_1, . . . , v_{k-1}, v_0]$ with k> 3 contains, which has no tendon. DenoteT_i the v_i corresponding tree. Then applies$T_i \cap T_j$ *6th*= ∅ exactly when i and j itself modulo k differ by at most 1. (In the following, all indices are implicitly modulok expected.)

Pick a point a_i the end $T_i \cap T_{i+1}$ for i = 0,. . . , k - 1.May be b_i the last common point of the unique paths of a_i to a_{i-1} and from a_i to a_{i+1}. These paths are inT_i or in T_{i+1}; b_i so lies in $T_i \cap T_{i+1}$. May be$P._{i+1}$ the easy path that b_i withb_{i+1} connects. Obviously,$P._i \subseteq T_i$. So it applies$P._i \cap P._j$ = ∅ If i and j modulok to ... more⋃differ as 1. Also applies$P._i \cap P._{i+1}$ = $\{b_i\}$ for i = 0,. . . , k - 1.Hence is
$_i$ $P._i$ a simple circle in T. This contradicts the definition of a tree.

(i) ⇒ (iii) The proof is done by induction on the size of G. We assume that the theorem applies to all graphs that have fewer nodes than G to have.

is G complete, is T a single knot and the statement trivial. isG not related to connected components G_1,. . . ,G_k, then by induction there exists for each of the graphs G_i a corresponding tree T_i, which contains the statements from (iii) Fulfills. We connect any node ofT_i with any node of T_{i+1} around the alleged tree, the property (iii) for G met, to receive.

In the following we consider the case that G is neither complete nor incoherent. May bea a simplicial knot of G and be A = {a}∪Adj (a). Obvious isA. a maximum clique of G.

May be U = {u ∈A | Adj (u) ⊂A} and Y = A - U. It applies a ∈U; see Figure 3.5. ThereG is not complete, there are knots in V - A, and there too G is connected, and nodes in U not to knot in V −A are adjacent can also Y not be empty. The quantitiesU, Y and V - A so are not empty.

We now consider the induced subgraph $G' = G_{V-U}$which is chordal and has fewer knots than G. According to the induction hypothesis, let T ′a tree whose set of nodes *K*′the amount of maximum cliques from G′is such that for each node in V − U the set of nodes $K'_v = \{X \in K' \mid v \in X\}$ a connected subgraph (Subtree!) Of T ′induced.

Comment. is Y a maximum clique of G′, then applies *K* = *K*′+ {A} - {Y}, otherwise applies *K* = *K*′+ {A}.

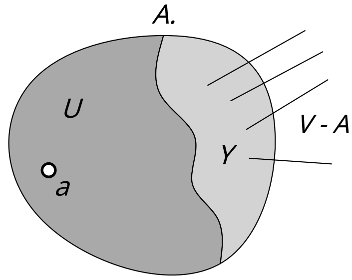


Figure 3.5: Illustration of the clique A, of their two subsets U and Y as well as her complement V - A

May be B. a maximum clique of G′, the Y contains. We differentiate between two cases, depending on whetherB = Y or not.

Case 1. is B = Y, so we get T the end T′ by making the knot B. in A. rename.

Case 2. is B. *6th*= Y, so we get T the end T′ by making the new node A. as a leaf the knot B. append.

In both cases it is K_u = {A} for all u ∈ U and $K_v = K'_v$ for all v ∈ V - A. These According to the assumption, sets each induce a subtree of T. So we have to just the quantities K_y with y ∈ Y check. In case 1 is $K_y = K'_y$ + {A} - {B}, What induces the same subtree as K'_y because we only renamed one node. In case 2 is $K_y = K'_y$ + {A}, which obviously induces a subtree. So that isT the searched tree and the theorem proved. □

3.4 Perfectness

We will now show that chordal graphs are perfect. For this we use the fact that node separators in chordal graphs form cliques (Lemma 3.4).

Theorem 3.15. *May be* S. *a separator of a connected undirected graph* G = (V, E) *and be* $G_{A_1}, \ldots, G_{A_t}$ *the connected components of* G_{V-S}. *is*S. *a clique, so is true*

$$\chi(G) = \max_i \chi(G_{S+A_i})$$

and

$$\omega(G) = \max_i \omega(G_{S+A_i}).$$

Proof. Obviously, $\chi(G) \geq \chi(G_{S+A_i})$ for all i. On the other hand, we can G_S. First of all, color as you like and the chosen color (in which each node has its own color) becomes a color for each of the G_{S+A_i} expand that are at most $\chi(G_{S+A_i})$ Colours used. Overall, this results in a color ofG with a maximum of $\max_i \chi(G_{S+A_i})$Colours.

Analogously, $\omega(G) \geq \omega(G_{S+A_i})$ for all i. On the other hand, no clique can X in GKnot off A_i and A_j with $j \neq i$ included as S. is a separator. Therefore every clique has toX completely in one of the subgraphs G_{S+A_i} be included, so equality follows. □

Corollary 3.16. *May be* S. *a separator of a connected graph* G = (V, E) *and be* $G_{A_1}, \ldots, G_{A_t}$ *the related components of* G_{V-S}. *is*S. *a clique and each subgraph* G_{S+A_i} *perfect so is* G *Perfect.*

Proof. We assume that the statement for all graphs that have fewer nodes than G, is applicable. Hence $\omega(G') = \chi(G')$ for all real induced subgraphs G' from Gand it suffices to show that $\omega(G) = \chi(G)$.

For each of the subgraphs we have $\chi(G_{S+A_i}) = \omega(G_{S+A_i})$ due to the perfection, so in particular $\max_i \chi(G_{S+A_i}) = \max_i \omega(G_{S+A_i})$. According to the preceding sentence, the following applies$\chi(G) = \omega(G)$. □

Theorem 3.17. *Chordal graphs are perfect.*

Proof. May be G a chordal graph. We show the statement by induction about the number of nodes and assume that the statement applies to all graphs that have fewer nodes thanG. We can assume without reservation that G coherent but not a clique. Then there is a separatorS. in G, the G in related components$G_{A_1}, \ldots, G_{A_t}$ disassembled with $t \geq 2$. Each of the graphs $G_{S+A_1}, \ldots, G_{S+A_t}$ is chordal and therefore perfect according to the induction hypothesis. According to Lemma 3.4S. a clique and so is also according to Corollary 3.16 G Perfect. □

3.5 Algorithms for Chordal Graphs

In the following we will use efficient algorithms for the problems Coloring, clique, independent set and Clique Cover specify on chordal graphs. The following is about thisG = (V, E) a chordal graph and σ a perfect elimination scheme for G.The first observation, going back to Fulkersson and Gross, is the fact that every maximal clique of the form $\{v\} \cup X_v$ is, where

$$X_v = \{x \in \mathrm{Adj}(v) \mid \sigma^{-1}(v) < \sigma^{-1}(x)\}.$$

On the one hand, each of the sets $\{v\} \cup X_v$ Completely. On the other hand isA. any maximal clique, we consider the first node w the end A. in the nodal order σ. Then appliesA = $\{w\} \cup X_w$. The following result in particular follows from this.

Proposition 3.18. *A chordal graph with* n *Node has at most* n *maximum inclusion cliques. Equality is present if and only if the graph has no edges.*

It is easy to modify algorithm 3 so that in each step the set {v} $\cup X_v$ is issued with. However, not all of these amounts are maximum. In order to decide which of these cliques are really maximal, it suffices to observe that the clique {v}$\cup X_v$ exactly then *not* is maximum if in a previous step all nodes in X_v at once to A (v) were added. (Proof: practice!)

Since σ is a perfect elimination scheme, in each step the set of is to A (v)added nodes are a subset of X_v. It is therefore sufficient to save the size of the largest amounts added in this way. By comparing this number with the size of$|X_v|$ we can then edit the node v decide whether {v} $\cup X_v$ is a maximum clique and must therefore be spent. Since chordal graphs are perfect, the size of the largest clique is also equal to the chromatic number. Algorithm 5 shows the pseudocode for this procedure.

```
procedure CLIQUES (σ)
Beginning
    χ ← 1;
    for all knots v do S (v) ← 0for i
    ← 1 until n do
        v ← σ (i);
        X ← {x ∈Adj (v) | σ−1(v) <σ−1(x)};if
        Adj (v) = ∅then print {v};if X = ∅
        then go to line 16;u ← σ (min {σ−1(x)
        | x∈X}); S (u)← Max{S (u), | X | - 1};
        if S (v) <| X | then

            print {v} ∪X;
            χ ← max {χ,1 + | X |};
        end
    end
    print "The chromatic number is" χ;
end
```

Algorithm 5: Procedure for listing all maximum cliques and calculating the chromatic number.

Theorem 3.19. *Algorithm 5 correctly calculates the chromatic number and all maximum cliques of a chordal graph* G = (V, E) *in* O (| V | + | E |) *Time.*

The proof can be carried out in a similar way to the proof of Theorem 3.11. (A practice!)

Next we deal with the computation of α (G). There G is perfect isα (G) = k (G), and at the same time we want to give an independent set and a clique coverage of this size.

To do this, we inductively define a sequence of nodes $y_1, y_2, \dots, y_t$ by $y_1 = \sigma(1)$

and $y_i = \sigma(\min\{\sigma^{-1}(v) \mid \sigma^{-1}(v) > \sigma^{-1}(y_i), v \notin X_{y_1} \cup X_{y_2} \cup \cdots \cup X_{y_{i-1}}\})$ as the first successor of y_i in the nodal order σ, which is not in any of the sets X_{y_j} with $j < i$ is included. In particular,

$$V = \{y_1, \ldots, y_t\} \cup X_{y_1} \cup \cdots \cup X_{y_t}.$$

The following sentence applies.

Theorem 3.20. *The amount* $\{y_1, \ldots, y_t\}$ *is a maximal independent set of* G *and the set of sets* $Y_i = \{y_i\} \cup X_{y_i}$ *(*$i = 1, 2, \ldots, t$*) is a minimal clique coverage of* G*.*

Proof. The amount $\{y_1, y_2, \ldots, y_t\}$ is independent there $y_j y_i \in E$. with $j < i$ implies that $y_i \in X_{y_j}$, what the definition of y_i contradict. So we have $\alpha(G) \geq t$. On the other hand is each of the sets $Y_i = \{y_i\} \cup X_{y_i}$ a clique, and therefore $\{Y_1, \ldots, Y_t\}$ a coverage of G with cliques. So we have $\alpha(G) = k(G) = t$, and we have indeed found a maximal independent set and minimal clique coverage. It is not difficult to implement the algorithm with linear running time. □

Chapter 4

Comparability graph

In this section we want to deal more closely with comparability graphs, i.e. with graphs that have a transitive orientation. We will see that these graphs are also perfect and provide an algorithm for recognizing the graphs.

4.1 Γ-chains and implication classes

An undirected graph G = (V, E) is a *Comparability graph* if an orientation(V, F) from G exists with

$$F. \cap F.^{-1} = \varnothing, F + F^{-1} = E, F^2 \subseteq F,$$

whereby $F.^2$ = {ac | ab, bc∈F. for a knot b}. The relation F. is a strike partial order of V, their comparability relation E. is and F. called *transitive orientation*from G (or from E).

For example, consider a circle with the nodes a, b, c, d (in that order along the circle); see figure 4.1. Any choiceaway ∈F. *enforces* the orientation of the other too b incidental edge too b there, so cb ∈F. Analogously, it also forces others to orientate themselves a incidental edge of a gone, well ad ∈F.Finally must too cb ∈F. are valid. We now want to clarify this concept of “forcing”.

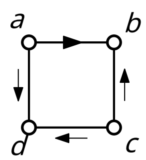


Figure 4.1: Example of forced edge directions.

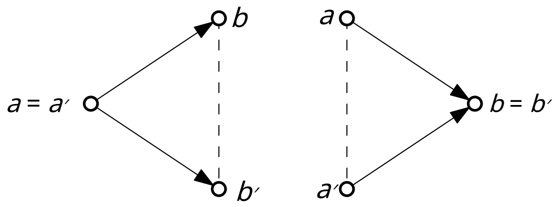


Figure 4.2: Illustration of the relation Γ.

We define a binary relation Γ on the edges of an undirected graph G = (V, E) as follows, see Figure 4.2:

$$\text{from } \Gamma\ ab' \quad \text{exactly when} \quad \begin{cases} \text{either} & a = a' \text{ and } bb' \notin E \\ \text{or} & b = b' \text{ and } aa' \notin E \end{cases}$$

if from Γ a b′ holds, we say that away the edge a b′ *directly enforces*. ThereE. is irreflexive, applies from Γ from, but away*6th* Γ ba. Obviously, the reflexive transitive closure is Γ_* fromΓ an equivalence relation R. and partitioned E. thus in equivalence classes that we *Implication classes* from G to name. Two edgesaway and CD are in the same implication class if and only if a sequence of edges exists

$$ab = a_0b_0\ \Gamma\ a_1b_1\ \Gamma\ \cdots\ \Gamma\ a_kb_k = cd \text{ with } k \geq 0.$$

Such a sequence is called Γ-*Chain* from away to CD. We say that away the edge CD *enforces*if from Γ_* CD.

The following properties apply (practice!).

$$\text{from } \Gamma\ ab' \Leftrightarrow ba\ \Gamma\ ba'$$

$$\text{from } \Gamma_*\ ab' \Leftrightarrow ba\ \Gamma_*\ ba'$$

May be I.(G) the set of implication classes of G. We define

$$\hat{I}(G) = \{\hat{A} \mid A. \in I(G)\},$$

where $\hat{A} = A. \cup A.^{-1}$ the symmetrical termination of A. designated. The elements of$\hat{I}$(G) mean *Color classes*.

Theorem 4.1. *May be* A. *an implication class of an undirected graph* G*. Owns*G *a transitive orientation* F.*, then either* $F. \cap \hat{A} = A$ *or* $F. \cap \hat{A} = A^{-1}$*. The following applies in both cases*$A. \cap A.^{-1} = \varnothing$*.*

Proof. The relation Γ is defined in such a way that for every transitive orientation F.from G is applicable:

$$\text{if} \quad \text{from } \Gamma\ ab' \quad \text{and} \quad \text{away} \in F. \quad \text{then} \quad ab' \in F.$$

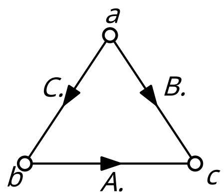


Figure 4.3: Requirement of the triangular lemma.

Repeated use of this property yields F. ∩ A = ∅ or A. ⊆ F. On the other hand, (i) A. ⊆ F + F_{-1} and (ii) F. ∩ $F._{-1}$ = ∅.

Is now F. ∩ A = ∅so follows

$$F. \cap A = \emptyset \Rightarrow A. \subseteq F._{-1} \text{ [because of (i)]} \Rightarrow A._{-1} \subseteq F. \Rightarrow \qquad F. \cap \hat{A} = A._{-1}.$$

Is however A. ⊆ F, so follows

$$A. \subseteq F. \Rightarrow A._{-1} \subseteq F._{-1} \qquad \Rightarrow \quad F. \cap A._{-1} = \emptyset \text{ [because of (ii)]} \Rightarrow \quad F. \cap \hat{A} = A.$$

The following applies in both cases A. ∩ $A._{-1}$ = ∅. □

We will see later that the converse of Theorem 4.1 also holds; is namelyA. ∩$A._{-1}$ = ∅ for each implication class of A, so owns G a transitive orientation.

Comment. Ma∪n could now conjecture that an arbitrary union of implication classes F = $_i$ $A._i$ with F. ∩ $F._{-1}$ = ∅ and F + F_{-1} = ∅ always a transitive order from G forms. However, this is not the case. For example, if you look at a triangle, you will see that it has exactly six implication classes, each of which contains a single edge in one direction. To get an orientation of the above design there is exactly2^3 = 8 Options. However, two of them are not transitive.

We will use the next two lemmas over and over again throughout the chapter. May beab = a_0b_0 Γ a_1b_1 Γ ··· Γ a_kb_k = cd a Γ chain. Fori = 1,. . . , kis applicable

$$a_{i-1}b_{i-1} \,\Gamma\, a_ib_{i-1} \,\Gamma\, a_ib_i,$$

because the added middle edge coincides with the first or second edge. We get the following statement.

Lemma 4.2. *Is applicable* from Γ∗CD*, there is one* Γ*Chain of* away *to* CD *the form*

$$ab = a_0b_0 \,\Gamma\, a_1b_0 \,\Gamma\, a_1b_1 \,\Gamma\, a_2b_1 \,\Gamma\, \cdots \,\Gamma\, a_kb_k = cd.$$

Such a chain is called *canonical* Γ*-Chain*. The use of canonical chains occasionally simplifies evidence.

Lemma 4.3 (Triangular lemma). *Be* ABC *Implication classes of an undirected graph* G = (V, E) *with* A. *6th*= B. *and* A. *6th*= C.$_{-1}$*, so that* away ∈ C, ac ∈ B. *and* bc ∈ A.*(Figure 4.3).*

(i) Is b c′ ∈ A.*, then applies* away′ ∈ C. *and* ac′ ∈ B..

(ii) Is b c′ ∈ A. *and* a b′ ∈ C.*, then* a c′ ∈ B..

(iii) No edge off A. *is to the knot* a *incidental.*

Proof. By Lemma 4.2 there is a canonical Γ-chain

$$bc = b_0c_0 \;\Gamma\; b_1c_0 \;\Gamma\; b_1c_1 \;\Gamma \ldots \Gamma\; b_kc_k = bc'.$$

We now inductively show the existence of away$_i$ ∈ C. and ac$_i$ ∈ B. for $0 \leq i \leq k$. For $i = 0$ is that obviously fulfilled. We now consider the case$i \geq 1$.

There ac$_{i-1}$ ∈ B, b$_i$c$_{i-1}$ ∈ A, but A. *6th*= B, is applicable away$_i$ ∈ E. There b$_i$c$_{i-1}$ Γ b$_{i-1}$c$_{i-1}$ follows that b$_i$b$_{i-1}$ ∉ E. It follows that away$_{i-1}$ and away$_i$ in the same implication class, namely C, lie.

There b$_i$c$_i$ ∈ A. and A. *6th*= C.$_{-1}$, follows b$_i$c$_i$*6th* Γ b$_i$a, so ac$_i$ ∈ E. There b$_i$c$_i$ Γ b$_i$c$_{i-1}$, are c$_i$ and c$_{i-1}$not adjacent. Hence appliesac$_i$ Γ ac$_{i-1}$, so ac$_{i-1}$ ∈ B. This proves property (i).

Furthermore, property (iii) follows directly from property (i). Is there an edgeb c′ ∈ A, then from (i) the existence of the edges follows away′and ac′. But now appliesb′= a or c′= a,so this contradicts the irreflectivity of E.

For (ii) we first assume that B. *6th*= C. Consider the canonical Γ-chain ab = a$_0$b$_0$ Γ a$_1$b$_0$ Γ a$_1$b$_1$ Γ a$_2$b$_1$ Γ ⋯ Γ a$_k$b$_k$ = a b′in C. We show by induction that for i = 0,. . . , kthe triangle a$_i$b$_i$c exists and is isomorphic to the triangle abc (i.e. the edges are in the same implication classes). Obviously, that is true of the trianglea$_0$b$_0$c = abc. Be now $i \geq 1$ and let's assume that the triangles a$_{i-1}$b$_{i-1}$c exists and is isomorphic to ABC. Then b$_{i-1}$a$_i$ ∈ C.$_{-1}$ and b$_{i-1}$c ∈ A. There A. *6th*= C.$_{-1}$, followsa$_i$c ∈ E. Also applies a$_i$c Γ a$_{i-1}$c ∈ B. Now we consider the pair of edges a$_i$b$_i$ ∈ C. anda$_i$c ∈ B. According to requirement is B. *6th*= C. That supplies the existence of the edge b$_i$c ∈ E.It applies b$_i$c Γ b$_{i-1}$c ∈ A. In particular, there is the last triangle of the sequence, a b c.Statement (i) then directly yields the existence of the edge a c′ ∈ B.

Let us now consider the remaining case B = C. The consideration of the pair of edgesb a′ ∈ B.$_{-1}$ and b c′ ∈ A. delivers because of A. *6th*= C.$_{-1}$ = B$_{-1}$ directly the existence of the edgea c′ ∈ E. We now assume that a c′ ∈ D. with C. *6th*= D. According to part (i), on the one hand, the following appliesac′ ∈ B = C. On the other hand, we can apply part (i) to the triangle as follows a b c′ apply with reversed edges; namely on the edgesc b′ ∈ A.$_{-1}$, b a′ ∈ C.$_{-1}$ and c a′ ∈ D.$_{-1}$. That works thereB.$_{-1}$ = C$_{-1}$ *6th*= D.$_{-1}$ and (A.$_{-1}$)$_{-1}$ = A *6th*= C.$_{-1}$ = B$_{-1}$. The existence of the edgeba ∈ B.$_{-1}$ then yields the edge according to part (i) c a ∈ D.$_{-1}$, soac′ ∈ D. *6th*= C. That is a contradiction. □

Theorem 4.4. *May be* A. *an implication class of an undirected graph* $G = (V, E)$*. Exactly one of the following statements applies:*

(i) $A = \hat{A} = A^{-1}$

(ii) $A. \cap A.^{-1} = \varnothing$*,* A. *and* $A.^{-1}$ *are transitive and the only possible transitive orientations of* $\hat{A}$*.*

Proof. (i) Adopted $A. \cap A.^{-1}$ *6th*$= \varnothing$. May beaway $\in A. \cap A.^{-1}$, so from Γ_*ba. For eachCD $\in$ A. is applicable cd Γ_* away and dc Γ_* ba. Since Γ_* is an equivalence relation, so it holds cd Γ_* dcand dc $\in$ A. So $A = \hat{A}$.

(ii) Adopted $A. \cap A.^{-1} = \varnothing$. Beab, bc $\in$ A. Were ac $\in$ / E, would be like that from Γ cb, so cb $\in$ A. and thus bc $\in A.^{-1}$; a contradiction. So it isac $\in$ E.

May be B. the implication class of G, the ac contains. AcceptedA. *6th*= B. We apply statement (i) of the triangle lemma to the edge away $\in$ A. and get with it away $\in$ B.So ac $\in$ A. and A. is transitive. It also follows that$A.^{-1}$ is transitive.

Also is A. an implication class of $\hat{A}$. According to Theorem 4.1 areA. and $A.^{-1}$ the only transitive orientations of $\hat{A}$. □

Corollary 4.5. *Any color class of an undirected graph* G *either has exactly two transitive orientations, one of which is the reverse of the other, or it has no transitive orientation at all. If*G *contains a color class that has no transitive orientation* G *no comparability graph.*

4.2 Algorithm for Finding Transitive Orientations

In this section we will develop an algorithm to decide whether a given graph is a comparability graph.

May be $G = (V, E)$ an undirected graph. A decomposition$E = \hat{B}_1 + \hat{B}_2 + \cdots + \hat{B}_k$ the Set of edges E. called G*-Disassembly* from E. if $B._i$ for each $i = 1,2,\ldots, k$an implication class of $\hat{B}_i + \cdots + \hat{B}_k$ is. A sequence of edges $[x_1y_1, x_2y_2, \ldots, x_ky_k]$ called*Disassembly scheme* from G when a G-Disassembly $E = \hat{B}_1 + \cdots \hat{B}_k$ exists with $x_iy_i \in B._i$for $i = 1,\ldots, k$. Obviously it can for anyone G-Decomposition give many different decomposition schemes (any representative set of the $\hat{B}_i$). On the other hand is aG-Decomposition clearly determined by specifying a decomposition scheme.

Obviously, a decomposition scheme and the associated G-Decomposition can easily be calculated using a greedy method. Algorithm 6 shows the procedure.

An important observation is that by removing the edges, implication classes may merge with one another. In the following we will describe in more detail how and under what circumstances this happens. Before that we need the concept of a multiplex.

May be G = (V, E) undirected graph.
Initiali = 1 and E_1 = E.

Step 1: Choose any edge $e_i = x_i y_i \in E_i$ Step 2: Count the implication class B_i from E_i on the $x_i y_i$ contains. Step 3: define $E_{i+1} = E_i - \hat{B}_i$. Step 4: if $E_{i+1} = \varnothing$, put k = i and stop. Otherwise increasei at 1

and go to step 1.

Algorithm 6: Decomposition algorithm

May be G = (V, E) an undirected graph. A complete subgraph (V_S., S) with r + 1Knot is called *simplex* with *rank* r if every undirected edge âb from S. to another color class of G heard. The one by a simplexS. with rank r generated subgraph(V_M., M) from G with M = {from $\in$ E | from Γ_*xy for a xy $\in$ S} called *Multiplex* with rank r.

Theorem 4.6. *May be* A. *an implication class of an undirected graph* G = (V, E) *and be*D. *an implication class of* E - Â. *Exactly one of the following statements applies.*

(i) D. *is an implication class of* E. *and* A. *is an implication class of* E - D̂.

(ii) D = B + C *whereby* B. *and* C. *Implication classes of* E. *are and* Â + B̂ + Ĉ *a multiplex of* E. *with rank 2.*

Proof. Removing Â from E. can cause some implication classes of E. merge with each other. May beD. the union of k Implication classes of
E.

We first consider the case that k $\geq$ 2; then there is a triangle of nodes ABC with bc $\in$ Â and either ac $\in$ B. and away $\in$ C. or but approx $\in$ B. and ba $\in$ C, whereby B. and C. different in D. contained implication classes of E. are. Without loss of generality, we assume thatac $\in$ B. and away $\in$ C. (The other case is the same for D^{-1}.) Accepted B = C^{-1}. Then would beba, ac $\in$ B, but bc $\notin$ B. According to Theorem 4.4 then applies B = B̂ = B^{-1}what implies that B = C, a contradiction. So B̂ holds$\cap$ Ĉ = $\varnothing$ and $G_{\{ABC\}}$ is indeed a three-colored triangle, so Â + B̂ + Ĉ a multiplex with rank 2.

Every Γ chain in E − Â, contains edges from B̂ and Ĉ, cannot contain edges from other implication classes, since all triangles in E. with an edge in Â and an edge in B̂ (or Ĉ) have the third side in Ĉ (or B̂) according to the triangular lemma. So it appliesk = 2 and D = B + C.

We now consider the case k = 1 and show that then A. an implication class ofE - D̂ is. From what we've already proven, we know ifA. no implication class of E - D̂ is, then D̂ + Â + $\hat{A}_1$ a multiplex of rank 2 in E. for a third implication class A_1 from E. But that contradicts the assumption that D.an implication class of E - Â is; a contradiction tok = 1. So is A. in fact an implication class of E - D̂.

□

Theorem 4.7 (TRO theorem). *May be* G = (V, E) *an undirected graph with* G*-Disassembly*E = $\hat{B}_1 + \cdots + \hat{B}_k$*. The following statements are equivalent:*

(i) F = $B_1 + B_2 + \cdots + B_k$ *is a transitive orientation of* G

(ii) G = (V, E) *is a comparability graph.*

(iii) $A \cap A^{-1} = \varnothing$ *for each implication class* A. *from* E..

(iv) $B_i \cap B_i^{-1} = \varnothing$ *for* i = 1,. . . , k*.*

(v) Any closed path of edges $v_1v_2, v_2v_3, \ldots, v_qv_1 \in E$*, so that* $v_{q-1}v_1, v_qv_2, v_{i-1}v_{i+1} \notin E$. *(for* i = 2,. . . , q - 1*) has a straight length.*

Proof. (i) ⇒ (ii) obviously applies F. transitive orientation of G.

(ii) ⇒ (iii) is just the statement of Theorem 4.1.

(iii) ⇒ (iv). We use complete induction. ThereB_1 an implication class of E. is, applies $B_1 \cap B_1^{-1} = \varnothing$. is$k = 1$, are we done. We now assume that the implication for all G-Decompositions of graphs of length less than k is applicable. In particular, it applies toE - $\hat{B}_1$.

May be D. an implication class of E − $\hat{B}_1$. By Theorem 4.6,D. either an implication class of E (then applies $D \cap D^{-1} = \varnothing$) or D = B + C, whereby B. and C. Implication classes of E. are such that $\hat{B} \cap \hat{C} = \varnothing$. The latter implies that

$$
\begin{aligned}
D \cap D^{-1} &= (B + C) \cap (B^{-1} + C^{-1}) \\
&= (B \cap B^{-1}) + (C \cap C^{-1}) = \varnothing.
\end{aligned}
$$

So inductively applies $B_i \cap B_i^{-1} = \varnothing$ for i = 2,. . . , k.

(iv) ⇒ (i). May be E = $\hat{B}_1 + \cdots + \hat{B}_k$ one G-Decomposition of E. with $B_i \cap B_i^{-1} = \varnothing$. According to Theorem 4.1B_1 transitive. is$k = 1$, so the implication holds. We now assume that the implication is for everyoneG-Decompositions of graphs with length less than k is applicable. According to this assumption isF = $B_2 + \cdots + B_k$ a transitive orientation of E - $\hat{B}_1$. It remains to be shown thatB_1 + F is transitive.

Be ab, bc $\in B_1$+ F. Are both edges in B_1 or both in F. so it follows from the transitivity of B_1 respectively. F, that too ac $\in B_1$ + F. So we assume that away $\in B_1$ and bc $\in$ F.(The case that away $\in$ F. and bc $\in B_1$ goes analogously.) According to the prerequisite, applies away *6th* Γ∗ cb,so ac $\in$ E. Accepted ac $\notin B_1$ + F. Then applies approx $\in B_1$ + F. the end approx $\in B_1$ and away $\in B_1$ follows by means of the transitivity of B_1, that cb $\in B_1$; a contradiction. Analogously it follows fromapprox $\in$ F. and bc $\in$ F, that ba $\in$ F; also a contradiction. Hence appliesac $\in B_1$ + F and so is F = $B_1 + \cdots + B_k$ a transitive orientation of E.

(ii) ⇔ (v). is G If there is no comparability graph, there is (according to statement (iii)) an implication class A. from E. with $A \cap A^{-1}$ *6th*= ∅. Accepted$v_1v_2 \in A \cap A^{-1}$. By Lemma 4.2

there is a Γ-chain

$$v_1v_2 \,\Gamma\, v_3v_2 \,\Gamma\, v_3v_{4th} \,\Gamma\, \cdots \,\Gamma\, v_qv_{q-1} \,\Gamma\, v_qv_{q+1} = v_2v_1 .$$

We then consider the closed path v_1v_2, v_2v_3,. . . , $v_{q-1}v_q$, v_qv_1. By construction isq odd, because the index of the first node in the chain is odd. The edge sequence thus forms exactly such a closed path.

Contains E. conversely, such a closed path with an odd length q, so applies

$$v_1v_2 \,\Gamma\, v_3v_2 \,\Gamma\, v_3v_{4th} \,\Gamma\, \cdots \,\Gamma\, v_qv_{q-1} \,\Gamma\, v_qv_1 \,\Gamma\, v_2v_1 .$$

It follows that for the implication class A, the v_1v_2 contains, A. $\cap$ A.$_{-1}$ *6th*= $\varnothing$. So isG (according to statement (iii)) no comparability graph. □

By combining the TRO theorem with the decomposition algorithm, we obtain an algorithm for recognizing comparability graphs. In addition, we can also calculate a transitive orientation, if it exists; see algorithm 7.

```
1 Beginning
2     Initialize: i ← 1; E.i ← E; F.← ∅Choose xiyi ∈ E.i any;
3     Count implication classB.i from E.i on the xiyi contains;
4th
5     if B.i ∩ B.−1  i = ∅ then
6th       gap B.i to F. added;
7th   otherwise
8th       output "G is not a comparability graph ";
9         STOP;
10    end
11    E.i + 1 ← E.i - B̂i;if E.i + 1
12th  = ∅ then
13th      k ← i;
14th      STOP;
15th  otherwise
16        i ← i + 1;
17th      go to line 3;
18th  end
19th end
```

Algorithm 7: TRO algorithm

It can be shown that the algorithm can be implemented with linear space and, at most, a quadratic time requirement.

Theorem 4.8. *Recognition of comparability graphs and the calculation of a transitive orientation can be done in* $O(\Delta \cdot |E|)$ *Time and with* $O(|V| + |E|)$ *Place to be resolved; referred to here*Δ *the maximum degree of knot.*

In fact, both problems can even be solved in linear running time. Interestingly, the problem of testing whether a given orientation is transitive is equivalent to the problem of multiplying two matrices. According to current knowledge, this requires a significantly longer running time.

4.3 Coloring and other problems on comparability graphs

For everyone *acyclic* (not necessarily transitive) orientation F. of an undirected graph G = (V, E) an associated strict partial order can be defined by x <y if and only if a non-trivial path of x to y exists. We now define one*Height function* H as follows: h (v) = 0 if v is a sink; otherwise ish (v) = 1 + Max{h (w) | vw∈F}. The function H can be calculated in linear time using recursive depth first search. (Exercise!) The functionH is a real knot coloring, but not necessarily a minimal one. The number of colors used is equal to the length of a longest path inF. An awkward choice of F. can lead to the use of many colors; isF. however transitive, the situation is more favorable.

Accepted G is a comparability graph and F. a transitive orientation of G.In this case, each path corresponds to in F. due to the transitivity of F. a clique of G. In this case, the height function yields a coloring with exactly ω (G) Colours. In addition, comparability graphs are hereditary, so ω ($G_{A.}$) = χ ($G_{A.}$) for each induced subgraph $G_{A.}$ from G. This shows the following sentence.

Theorem 4.9. *Comparability graphs are perfect.*

The following famous theorem is a direct consequence of this and the theorem about perfect graphs (Theorem 2.8).

Theorem 4.10 (Dilworth theorem). *May be* (X,≤) *a partially ordered set. The minimum number of linearly ordered subsets (*Chains*) around* X *to be covered is equal to the maximum size of a subset of* X*of which no two elements are comparable (* Anti chain *).*

With the algorithm from Section 4.2 we can move into a transitive orientation inO (Δ | E | + | V |) Calculate steps. Using the height function, we can then useO (| V | + | E |) Time to gain a minimal coloring. By finding a longest path, we can also calculate a maximum clique.

Finally we show the calculation of α (G). May be (V, F) a transitive orientation of G. We transform this into a flow network with two additional nodes s and t and the edges sx and yt for each source x and every sink y from F.In addition, we assign a lower bound of 1 for the capacity to each node. Froms outbound edges also have a cost of 1 for each unit of flow carried.

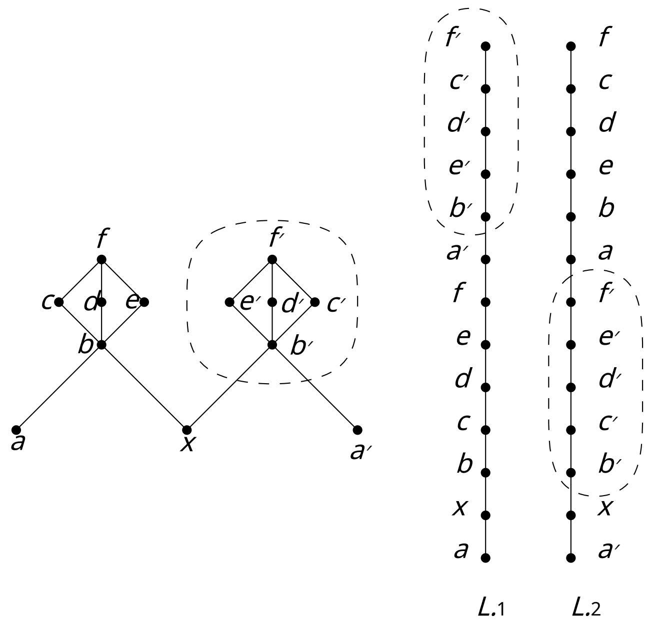


Figure 4.4: Example of partial order P. of dimension 2 and an associated minimal realizer of size 2.

The resulting flow problem can easily be described as a flow problem with minimal costs and can be solved in polynomial time. Such a flow can always be divided into a set ofst-Break down paths. Each of these paths then corresponds to a clique, so that we have an overall coverage of cliques. The minimization of costs ensures that a minimal clique coverage is chosen. Since comparability graphs are perfect, we have α (G) = k (G), and we thus have the value α (G) calculated.

4.4 The dimension of partial orders

May be (X, P) a partial order. We follow the English spelling and designate(X, P) also as *poset* (partially ordered set). Every partial order can be converted into a linear order by means of topological sortingL. from X expand. S.$\cap$ egg *L.*(P) the Set of all linear extensions of P. Any subset $L \subseteq L$($\cap$ P), the L.$\in L$ L = P fulfilled, is called *Implementer* from P; its size is $|L.|$. It isaway $\in$ L.$\in L$ L. exactly then if away $\in$ L. for each L. $\in L$.

Obvious is *L.*(P) himself a realizer of P. We define the dimension of P,dimP. as the smallest possible size of a realizer for P. Such a realizer is also called *minimal realizer* for P.

Lemma 4.11. *May be* (X, P) *a pose. For each*Y $\subseteq$ X *is applicable* dimP.$_Y \leq$ dimP..

Proof. Obviously, the limitation of a realizer results *L.* from P. on the elements of Y a realizer of P_Y. The statement follows by *L.* as the minimal realizer of P. chooses.

□

Theorem 4.12. *May be* G *the comparability graph of a pose* P.. *Then applies* dim P. ≤ 2 *if and only if the complement graph* $\overline{G}$ *is transitively orientable.*

Proof. May be F. a transitive orientation of $\overline{G}$. It's not hard to see that {P + F, P + F^{-1}} a realizer of P. is. Be the other way around {L_1, L_2} a realizer of P.Then F = L_1 - P = (L_2 - P)$^{-1}$ a transitive orientation of $\overline{G}$. That would be ab, bc $\in$ F. but ac $\notin$ F, so implies the transitivity of L_1, that ac $\in$ P. and the Transitivity of L_2, that approx $\in$ P; a contradiction. □

It follows directly from the previous sentence that two partial orders that have the same comparability graph either have both dimensions at most 2 or both dimensions are genuinely greater than 2. Even a stronger statement applies.

Theorem 4.13 (without proof). *Are* P. *and* Q *two orders with the same comparability graph* G*, then applies* dim P = dim Q*.*

Chapter 5

Split graph

An undirected graph G may have one or more of the following properties:

characteristic V: G is a comparability graph.characteristic V: G is a comparability graph ($\overline{G}$ is a *co-comparability graph*).characteristic C: G is chordal.characteristic C: G is chordal ($\overline{G}$ is *co-chordal*).

In fact, these four properties are independent of each other. There are sample graphs for each of the 16 possible combinations. (Exercise !?) In the following three chapters we will take a closer look at three of these combinations.

5.1 Characterization of split graphs

An undirected graph G = (V, E) is a *Split graph* if there is a decomposition V = S + K there so S. an independent set and K is a full set; see figure 5.1. The edges between S. and K are not subject to any restrictions. In general, the decompositionV = S + K of a split graph is neither unique nor isS. respectively. K necessarily a maximum independent set or maximum clique.

Since the complement of an independent set is complete and vice versa, the following theorem follows directly.

Theorem 5.1. *An undirected graph* G = (V, E) *is a split graph if and only if its complement* $\overline{G}$ *is a split graph.*

Theorem 5.2. *May be* G *a split graph whose nodes are in an independent set* S. *and a full set* K *were disassembled. Then exactly one of the following statements applies:*

(i) $|S| = \alpha(G)$ *and* $|K| = \omega(G)$

(in this case the partition is S + K *clearly)*

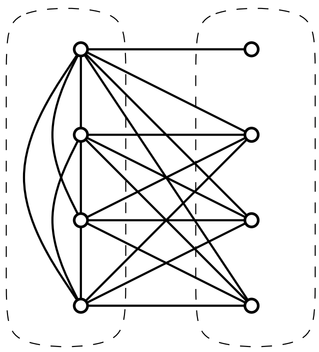


Figure 5.1: A split graph with one of its four possible partitions in S. and K. By moving one of the Grade 4 nodes out S. after K one again obtains a valid decomposition.

(ii) $|S| = \alpha(G)$ *and* $|K| = \omega(G) - 1$ *(in this case a* $x \in S.$*, so that* $K + \{x\}$ *is complete)*

(iii) $|S| = \alpha(G) - 1$ *and* $|K| = \omega(G)$ *(in this case a* $y \in K$*, so that* $S + \{y\}$ *is independent)*

Proof. Since an independent and a complete set share at most one node, the sum $\alpha(G) + \omega(G)$ either equal to $|V|$ or equal to $|V| + 1$.

If $\alpha(G) + \omega(G) = |V|$, then there is case (i). Suppose there is another partition $V = S' + K'$. May be $\{x\} = S \cap K'$ and $\{y\} = S' \cap K$; the sections are not empty, otherwise $S = S'$ and $K = K'$ would follow. In addition, such a section can of course contain at most one element. In addition, by definition, $x \in S.$ and $y \in K$, so x *6th*= y.Hence applies $K' = K - \{y\} + \{x\}$ and $S.' = S - \{x\} + \{y\}$. Are x and y adjacently so is $\{x\} + K$ a bigger clique, otherwise $\{y\} + S$ a larger independent crowd; a contradiction. So is the decomposition $V = S + K$ clearly.

If $\alpha(G) + \omega(G) = |V| + 1$, then there is case (ii) or (iii). We only prove case (ii), case (iii) works analogously. Let $|S| = \alpha(G)$, $|K| = \omega(G) - 1$ and be K' a clique of greatness $\omega(G)$. There $S + K$ is a partition and K' is greater than K, got to $S. \cap K'$ be non-empty and therefore have size 1. May be $\{x\} = S \cap K'$. Hence is $K' = K + \{x\}$ Completely. □

Theorem 5.3. *May be* G *an undirected graph. The following statements are equivalent:*

(i) G *is a split graph,*

(ii) G *and* $\overline{G}$ *are chordal,*

(iii) G *does not contain any induced subgraph isomorphic to* $2K_2$*,* $C._{4th}$ *or* $C._5$*.*

Proof. (i) ⇒ (ii). May be$G = (V, E)$ a split graph with decomposition $V = S + K$ as above. Accepted$G$ contains an induced chordless circle C. of length at least 4. Then there are at least one, but at most two of the nodes of C. in K contain. In the latter case the two knots must be open$C$. be adjacent. In both cases, however, contains$S$. a pair of nodes connected by an edge. Hence is$\overline{G}$ chordal. By Theorem 5.1 is also$\overline{G}$ a split graph and must therefore also be chordal.

(ii) ⇒ (iii). Obviously areC_4th and C_5 not chordal and $2K_2$ because $2K_2 = \overline{C_4}$th not co-chordal.

(iii) ⇒ (i). May be$K$ a (cardinality) maximum clique of G, which has been selected among all maximum cliques so that G_{V-K} has as few edges as possible. We have to show that$S = V - K$ is independent.

Accepted G_S. contains an edge xy. Due to the maximality of K can not knot from S. to each node K be adjacent. Would be$x$ and y both to all nodes K adjacent except for the same node z, would be like that $K- \{z\} + \{x\} + \{y\}$a greater complete amount than K. Hence there are two different nodes $u, v \in K$ with $xu \notin E$ and $yv \notin E$.

There G neither an induced copy of $2K_2$ another induced copy of C_4th contains, it follows that exactly one of the edges xv and yu in G is. Without loss of generality, we assume that$xv \notin E$ and $yu \in E$. Be now w any node of $K- \{u, v\}$. Were w to none of the nodes x, y neighboring so would be $G_{\{x, y, v, w\}} \cong 2K_2$. Were w not to y but adjacent to x, would be like that $G_{\{x, y, u, w\}} \cong C_4$th. Hence is$y$ to each node of $K - \{v\}$ adjacent and $K' = K - \{v\} + \{y\}$ is a maximum cardinality clique.

There $G_{V-K'}$no fewer edges than G_{V-K} can have follows from the fact that x to$y$ but not too v adjacent is that there is a knot t *6th*$= y$ in $V - K$ who has to admit v but not too y is adjacent. The edgetx must be in E. otherwise would $\{t, x, y, v\}$a copy of $2K_2$ induce. The same applies analogously$tu \notin E$, otherwise would $\{t, x, y, u\}$ a copy of C_4th induce. Then, however, $\{t, x, y, u, v\}$ a copy of C_5; a contradiction. That is the case$S = V - K$ independent and G a split graph. □

5.2 Degree sequences and split graphs

One episode $\Delta = [d_1, d_2, \ldots, d_n]$ of whole numbers with $n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$ called *graphically* if there is an undirected graph that Δ has as a degree sequence. For example, [2,2,2,2] graphically, as it is the degree sequence of C_4th is. Indeed it isC_4th the only graph with this degree sequence. The sequence [2,2,2,2,2,2] however, corresponds to both $2K_3$ as well as C_6th. It is easy to indicate episodes that are not graphically ind; approximately [1,1,1] and [4,4,2,1,1]. A simple necessary requirement is that s$\sum d_i$ must be straight. The previous example shows, however, that this is not sufficient.

We now give two classical characterizations of graphic sequences without proof at.

Theorem 5.4 (Havel '65, Hakimi '62). *One episode* Δ *of whole numbers with* $n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$ *is graphical if and only if the modified (not sorted in ascending order) sequence*

$$\Delta' = [d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n]$$

is graphic.

Theorem 5.5 (Erdős, Gallai '60). *A sequence of whole numbers* $n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$ *is graphic if and only if*

(i) $\sum_{i=1}^{n} d_i$ *straight, and*

(ii) $\sum_{i=1}^{r} d_i \leq r(r-1) + \sum_{i=r+1}^{n} \min\{r, d_i\}$ *for* $r = 1.2, \ldots, n - 1$.

Inequality (ii) is called r-te Erdős-Gallai inequality (EGU). Both sets imply efficient algorithms for testing whether a given sequence is graphical or not. (A practice!)

As we saw earlier, a degree sequence does not generally uniquely describe a graph. Therefore the question arises which graph properties are completely determined by the degree sequence. For example, it is relatively easy to show that transitively oriented complete graphs are completely characterized by the sequence of the input degrees. There is also a characterization of the degree sequences of trees. (Exercise!) In the following we will examine this question for split graphs.

May be $\Delta = [d_1, \ldots, d_n]$ a sequence of whole numbers with $n - 1 \geq d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$,and let $\zeta = [0,1,2, \ldots, n - 1]$.We now consider the position i the consequences of $\Delta$the sequence ζ overtakes. May bem the largest index i, so that $d_i \geq i - 1$. It is either $m = n$,then Δ the degree sequence of K_n, or $d_m \geq m - 1$ and $d_{m+1} < m$.

Split graphs can be characterized as those graphs for the m-te EGU is fulfilled with equality, where m is as previously defined. Intuitively represents the positionm just the transition in the list from the degrees of the clique nodes to the degrees of the nodes from the independent set.

Theorem 5.6. *May be* $G = (V, E)$ *an undirected graph with degree sequence* $d_1 \geq d_2 \geq \cdots \geq d_n$*, and be* $m = \mathrm{Max}\{i \mid d_i \geq i - 1\}$*. Then*G *a split graph if and only if*

$$\sum_{i=1}^{m} d_i = m(m-1) + \sum_{i=m+1}^{n} d_i .$$

Then applies furthermore $\omega(G) = m$.

Proof. The sentence certainly applies if G is complete. So we can assume that $d_m \geq m - 1$ and $d_{m+1} < m$. There Δ is not sorted in ascending order, min $\{m, d_i\} = d_i$ for $i \geq m + 1$. Therefore, the simplifies m-te EGU too

$$s = \sum_{i=1}^{m} d_i \leq m(m-1) + \sum_{i=m+1}^{n} d_i . \tag{5.1}$$

May be K the amount of the first m Knot of maximum degree. The left term of equation (5.1) can be divided into two contributions $s = s_1 + s_2$ disassemble with

$$s_1 = \sum_{x \in K} |\{z \in K \mid xz \in E\}| \leq m(m-1) \tag{5.2}$$

$$s_2 = \sum_{x \in K} |\, y \notin K \mid xy \in E\}|$$

$$= \sum_{y \notin K} |\{x \in K \mid xy \in E\}| \leq \sum_{i=m+1}^{n} d_i . \tag{5.3}$$

In inequality (5.2) equality holds if and only if K is complete. In inequality (5.3), on the other hand, equality applies if and only if $V - K$ is an independent set. Hence is G a split graph if inequality (5.1) is satisfied with equality.

The opposite is true $G = (V, E)$ a split graph. According to Theorem 5.2 we can V into an independent set S. and a full set K decompose so that $|K| = \omega(G)$. Every knot in K has degree at least $|K| - 1$, and since K cardinality is maximum, all nodes have S. at most degrees $|K| - 1$. So we can assume that the nodes are ordered in such a way that $K = \{v_1, \ldots, v_{|K|}\}$ and $S = \{v_{|K|+1}, \ldots, v_n\}$, where $\deg(v_i) = d_i$ is applicable. Also applies $d_{|K|} \geq |K| - 1$ and $d_{|K|+1} \leq |K| - 1 < |K|$, so $\omega(G) = |K| = m$. There K complete and $S = V - K$ are independent, the inequalities (5.2) and (5.3), and therefore also inequality (5.1), are fulfilled with equality. □

Corollary 5.7. *is* G *a split graph, every graph with the same degree sequence is also a split graph.*

Chapter 6

Permutation graph

In this chapter we consider a very basic class of perfect graphs with many uses. Let π be a permutation of the numbers $1,2,\ldots,n$. We look at π as a result $[\pi_1, \pi_2, \ldots, \pi_n]$; where π_i denotes the image $\pi(i)$. Analogously there are $(\pi^{-1})_i$, which we short as π^{-1}_i write the position of π_i in the sequence.

Given such a permutation π we can create an undirected graph $G[\pi]$ define as follows. The nodes of $G[\pi]$ are from 1 until n numbered and two nodes are connected to each other if the node with the larger number is in π to the left of the node with the smaller number (i.e. they were swapped by the permutation). The graph $G[\pi]$ also known as *Inversion graph*.

Formal can be $G[\pi]$ define as follows. Π is a permutation of numbers $1,2,\ldots,n$, so is the inversion graph $G[\pi] = (V, E)$ defined as follows:

$$V = \{1,2,\ldots,n\}$$

and

$$ij \in E. \Leftrightarrow (i - j)(\pi^{-1}_i - \pi^{-1}_j) < 0.$$

An undirected graph G is a *Permutation graph* if a permutation π with $G \cong G[\pi]$ exists. Figure 6.1 shows an example of a permutation graph.

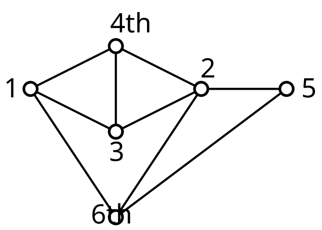


Figure 6.1: The permutation graph G [4,3,6,1,5,2]

6.1 characterization

Permutation graphs have a number of interesting properties. For example, consider what happens when we reverse the permutation π. Then every pair of numbers that was previously in the correct order in π is now in the wrong order and vice versa. So the permutation graph obtained is the complement graph ofG [π]. So denotes π_ρ the permutation obtained by reversing the sequence π holds

$$G[\pi_\rho] \cong \overline{G[\pi]}.$$

The complement of a permutation graph is also a permutation graph.

Another property of graphs G [π] is that they are transitively orientable. We get a transitive orientationF, by directing each edge towards the end point with the larger number. Are nowij $\in$F. and jk $\in$F, so applies i <j <k and $\pi^{-1}_i > \pi^{-1}_j > \pi^{-1}_k$, so ik $\in$F. In fact, the following stronger statement is true.

Theorem 6.1. *An undirected graph* G *is a permutation graph if and only if* G*and* $\overline{G}$ *Are comparability graphs.*

Proof. Accepted $G \cong G[\pi]$; then G Comparability graph, there G is transitively orientable. In addition, is$\overline{G} \cong G[\pi_\rho]$ also a permutation graph and thus also a comparability graph.

Conversely are (V, F_1) and (V, F_2) transitive orientations of G = (V, E) and $\overline{G}$ = (V, $\overline{E}$). We claim that (V, $F_1 + F_2$) an acyclic orientation of the complete graph (V, E + $\overline{E}$) is. AcceptedF.$_1$ + F_2 would contain a circle [v_0, v_1, v_2,. . . , v_{l-1}, v_0]with the smallest length l. is l> 3, so can the circle with v_0v_2 or v_2v_0 shorten what contradicts minimalism. Is howeverl = 3, so at least two of the edges of the circle are in the same F.$_i$what implies that F.$_i$ is not transitive. So is (V, $F_1 + F_2$) acyclic. Analogously it can be shown that (V, $F^{-1}_1 + F_2$) is acyclic.

We now construct a permutation π such that $G \cong G[\pi]$. An acyclic orientation of a complete graph is always transitive and determines a clear linear order of the nodes (by means of topological sorting). We now proceed in three steps:

Step I: Label the nodes using the F.$_1$ + F_2 certain order; the node x with degree of entry i - 1 receives the label L (x) = i.

Step II: Label the nodes using the F.$^{-1}_1$ + F_2 established order; the node x with degree of entry i - 1 receives the label L.(x) = i.

Note that

$$xy \in E. \Leftrightarrow [L(x) - L(y)]\,[L(x) - L.(y)] < 0, \tag{6.1}$$

because exactly the edges in E. reversed between step I and step II.

Step III: Define π as follows: for nodes x may be L (x) = i, then put π^{-1}_i = L. (x).
The permutation π thus transfers the labeling L. in the label L.´.

According to equation (6.1), π is the desired permutation and L. the desired graph isomorphism. □

Comment. With the terms from Section 4.4 is G a permutation graph if and only if the transitive orientations of G viewed as partial orders have at most dimension 2. The two labels constructed aboveL. and L.´
form a realizer of a transitive orientation of G.

Theorem 6.1 also describes an algorithmic procedure for the recognition of permutation graphs by creating a transitive orientation of G and $\bar{G}$ calculated. The procedure described in the proof then gives a permutation π. This method is required for graphs withn node O (n^3) time and O (n^2) Place.

From the transitive orientability there also follows an important connection between the permutation π and the cliques or independent sets of the graphG [π].

Comment. The descending subsequences of π just correspond to the cliques ofG [π]. The increasing subsequences of π just correspond to the independent sets of G [π].

A related, but slightly easier, problem is whether for a given numbering L.the node of a graph G with the numbers 1, . . . , na permutation π exists such thatG = G [π]. In this case the given numbering is called L. *Permutation labeling*. A characterization of the permutation labels is presented in the exercise.

6.2 Applications

Permutation graphs can be viewed as a class of intersection graphs in the following way. Write the numbers1,2,. . . , nhorizontally from left to right in one line. Below that, write the sequence of numbers π_1, π_2,. . . , π_n also from left to right. Finally, draw straight lines that will take the two of them1en, the two 2en etc. connect with each other. We also call this*Matching representation* of π; see picture. Note that the i-te and that j-intersect the th segment if and only if the sequence of i and j is swapped into π; this is the same criterion as the existence of an edge inG [π]. Therefore, the intersection graphs of segments of a matching diagram are exactly the permutation graphs.

Application 1

Suppose we have two sets X and Y of cities that lie on two parallel straight lines. Let us further assume that some of the cities inX with some

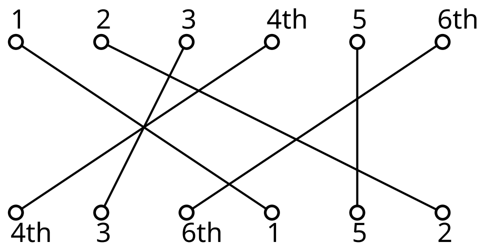


Figure 6.2: Matching representation of G [4,3,6,1,5,2].

of cities in Y connect, which should all be served at the same time. Our task is to determine an altitude for each of the flight routes so that intersecting routes have different altitudes. This ensures that collisions do not occur during flights. Obviously this is a staining problem.

The data is available as a bipartite graph embedded in the plane, as in Figure 6.3a. We first number the flight routes by going through the cities on the northern straight line from west to east. This gives us a matching representation and thus a corresponding permutation graphG [π]; see Figure 6.3b. Since assigning the flight levels is now equivalent to coloring the nodes ofG [π], so that neighboring nodes are given different colors; Figure 6.3c. ThereG [π] is a comparability graph, this can be done, for example, with the algorithm from Section 4.3. In the next section we will introduce a more efficient coloring algorithm for permutation graphs.

Application 2

May be $\mathcal{I}.$ = {$I._1$ | i = 1,. . . , n}a set of intervals of a straight line, where $I._i = (x_i, y_i)$and | $I._i$| =$y_i - x_i$ the length of $I._i$ designated. We further assume that the intervals that may overlap are sorted so that$x_1 \leq x_2 \leq \cdots \leq x_n$. Be$w_i$ the cost of moving the interval $I._i$ (we assume that the displacement cost is independent of the displacement distance). Find the most favorable shift of intervals so that (1) the order is retained and (2) no more intervals overlap after the shift. (The intervals could be, for example, the memory requirements of programs at certain times.)

A solution could look like this. We consider the oriented graph ($\mathcal{I}.$, F)with

$$(I._i, I._j) \in F. \quad \Leftrightarrow \quad \sum_{i \leq k \leq j} | I_k | \leq y_j - x_i \qquad (i < j).$$

Then two intervals are through if and only then F. connected with each other if the intervals between them can be shifted so that none of these can be shifted j - i + 1 Intervals

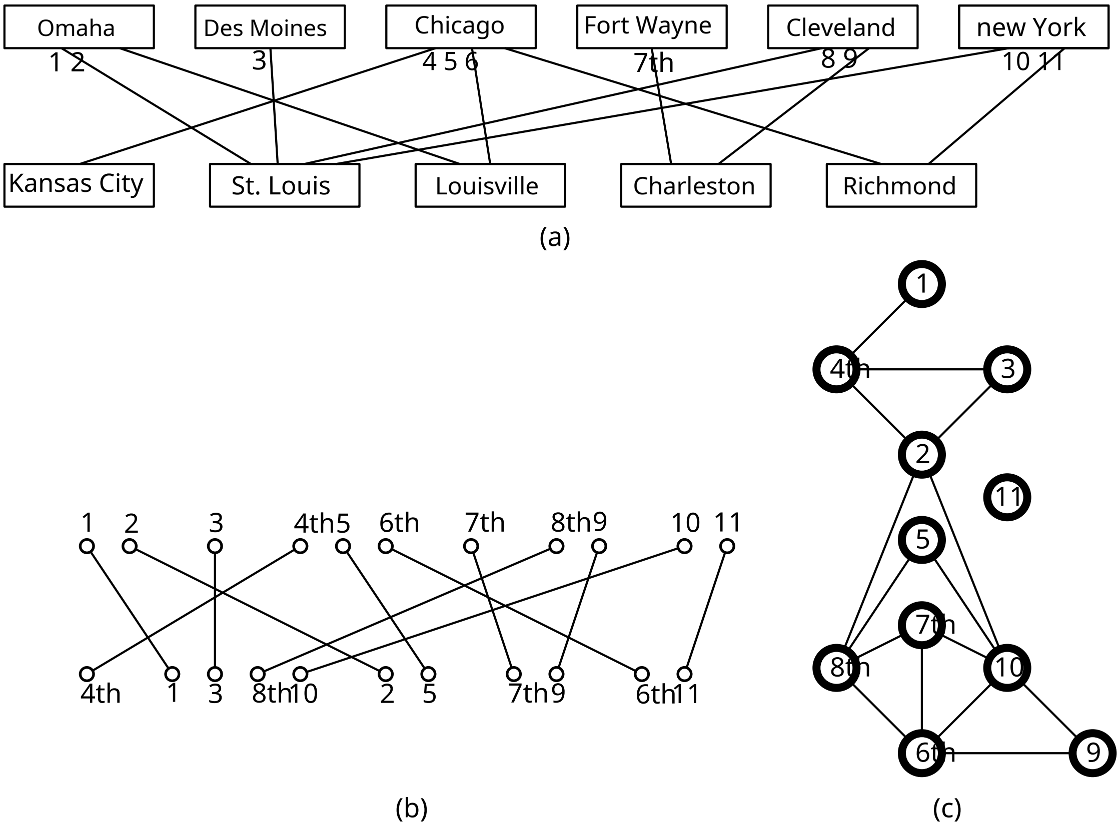


Figure 6.3: (a) A bipartite graph B, which represents flight routes between cities. (b) The matching representation of the from the bipartite graphB. constructed permutation π. (c) The graphG [π]. Coloring the knot of G [π] solves the altitude assignment problem for B.

(including the captured I_i and I_j) overlap. It's not difficult to show thatF.is transitive. The solution to the problem is now to put a chain inF. with maximum weight (to hold them in place, all other intervals are shifted). In other words, a clique with maximum weight in the graph (I., $F + F^{-1}$)being found. This graph is not only a comparability graph but even a permutation graph.

6.3 Sorting permutations with queues

A queue is a simple data structure in which data is entered at one end and taken from the other end according to the FIFO principle (first-in-first-out). We consider the problem of a permutation π of the numbers1,2,. . . , nwith the help of a network of k parallel queues as in

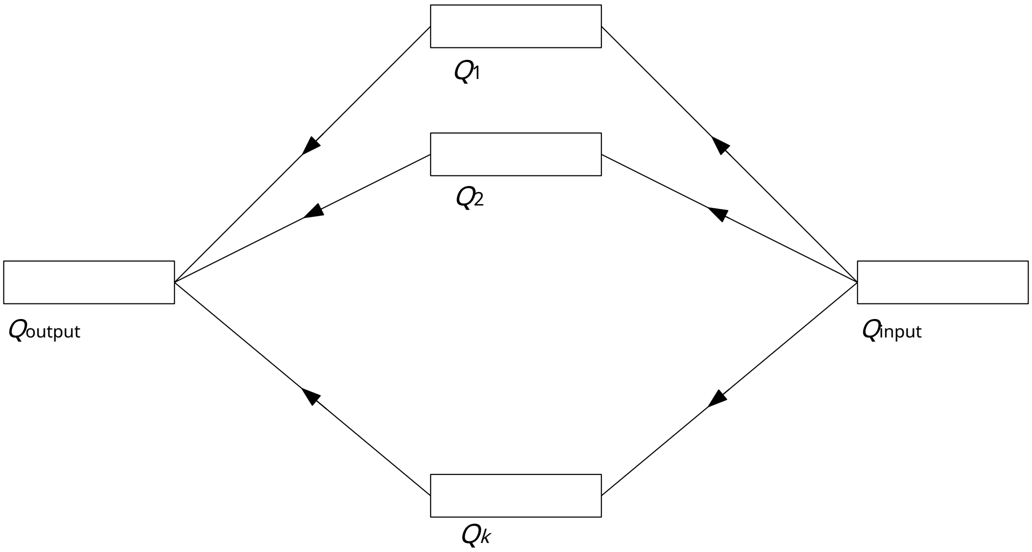


Figure 6.4: A network from k parallel queues. (Attention: data flow from right to left!)

Figure 6.4 to sort. At the beginning, the entire permutation is in the input queue. Little by little the numbers migrate from the input to thek inner queues, where they are temporarily stored until they are then moved to the output queue. We assume that each of the queues has unlimited capacity and that data cannot be moved against the direction of the arrow. You can imagine it like a marshalling yard where wagons are to be rearranged. The number of tracks that can be used in parallel (i.e. the number of queues) is often limited. This leads to the following question. Given a network ofk parallel queues, characterize the permutations that can be sorted with it. Or, given a permutation π, how many queues are needed to sort them? In addition, an optimal sorting should be found in this case.

The central question is: when do two numbers have to go through different queues? Exactly when they are exchanged in π, i.e. in the wrong order. If soi and j in G [π] are adjacent, they have to go through different queues.

Proposition 6.2. *May be* $\pi = [\pi_1, \ldots, \pi_n]$ *a permutation of numbers* $\{1,2,\ldots,n\}$*. There is a bijection between the real ones*k- *staining of* G [π] *and the successful sorting strategies for* π *in a network with* k *Queues.*

Proof. We assign everyone to queue Q_i another color too. Now we sort the permutation with thek-Network and color each number with the appropriate color when entering a queue. There inG [π] connected nodes i and j different

When passing through the queues, they are also given different colors.

Conversely, consider a true coloration of G [π] with colors 1,2,. . . , k.If the element x the input color c has, so we send c in the queue Q_c. Suppose this strategy is unsuccessful. Then one of the queues contains, sayQ_ma pair of numbers x and y in the wrong order. Than arex and y but in G [π]adjacent and yet colored with the same color, a contradiction. Obviously, this correspondence is a bijection.

□

Corollary 6.3. *May be* π *a permutation. The following numbers are the same.*

(i) The chromatic number of G [π]*,*

(ii) the minimum number of queues around π *to sort,*

(iii) the length of the longest descending subsequence in π*.*

Proof. The equivalence of (i) and (ii) follows directly from Proposition 6.2; the proof also shows how the corresponding solutions can be transformed into one another.

The equality of (i) and (iii) holds because such a longest subsequence of π is a maximum clique of G [π] whose size is just χ (G [π]) is that permutation graphs are perfect. □

the *canonical sorting strategy* for π inserts each number into the first available queue. This gives us a*canonical coloring* from G [π]. Algorithm 8 simulates this procedure. He calculates a minimal coloring. Using a binary search for the implementation of line 4 achieves a total runtime of$O$ (n logn).

Theorem 6.4. *May be* π *a permutation of numbers* {1,2,. . . , n}*. The canonical coloring of*G [π]*as computed by Algorithm 8 is minimal coloring.*

Proof. Obviously, Algorithm 8 computes a true χ-coloring of G [π]. We have to show that χ = χ (G [π]). It suffices to show that π contains a descending subsequence of length χ.

We consider the predecessor function defined as follows v: If COLOR (π_j) = i ≥ 2,so denote $\pi_{v(j)}$ the entry LAST (i - 1) during the j-th iteration. Obviously, π holds$_{v(j)} > \pi_j$ and v (j) <j, since the entry $\pi_{v(j)}$ At the end of Q_{i-1} stood and thus $\pi_j$in Q_i has forced.

Now let π_{j_χ}a knot with color χ. Then the result is

$$\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_\chi}$$

with $\pi_{j_{i-1}} = \pi_{v(j_i)}$ for i = χ, χ - 1,. . . , 2the desired descending order. □

Input : Permutation $\pi = [\pi_1, \ldots, \pi_n]$ of numbers $\{1, \ldots, n\}$.Output : Coloring of the nodes of $G[\pi]$ and chromatic number χ of $G[\pi]$.Proceed : In the j-th iteration becomes π_j in the queue Q_i with the
smallest index i shifted, for which it holds that π_j is larger than the last entry in Q_i. Instead of the entire queueQ_i to save, only an entry LAST (i) which is the last (largest) number in Q_i saves. The counterk saves the number of queues actually used.

```
Beginning
    k ← 0;
    for j ← 1 until n do
        i ← Index of the first permitted queue; COLOR
        (πj) ← i;LOAD(i) ← πj;k ← Max{k, i};


    end
    χ ← k;
end
```

Algorithm 8: Algorithm for canonical coloring of permutations.

To achieve a minimum clique coverage of $G[\pi]$ we can simply find algorithm 8 on the inverse π_ρ of π apply. For both problems, the runtime of the algorithm is in$O$ (n logn), provided the permutation π and the isomorphism $G \to G[\pi]$ are known. Interestingly, this is independent of the number of edges inG. However, if the figures are not known, we can either calculate them or use the coloring algorithm for comparability graphs from Section 4.3.

Chapter 7

Interval graphs

Interval graphs were historically one of the first types of intersection graphs. The question about the recognition of slice graphs was asked by G. Hajös as early as 1957. We already got to know interval graphs in chapter 1.3 and proved some basic properties there. In this chapter, we want to use the techniques from the previous chapters to gain a more detailed understanding of interval graphs and related graph classes.

7.1 Characterization

Theorem 7.1. *For an undirected graph* G *the following statements are equivalent.*

(i) G *is an interval graph.*

(ii) G *does not contain sinews* 4th*-Circle and its complement* $\overline{G}$ *is a comparability graph.*

(iii) The (inclusion) maximum cliques of G *can be ordered linearly so that for each node* x *from* G *the maximum cliques of* G*, the* x *contained in the order are consecutive.*

Proof. (i) ⇒ (ii): That is the statement of Proposition 1.6 and Proposition 1.8 from Section 1.3.

(ii) ⇒ (iii): We assume that $G = (V, E)$ not a sinewless 4-Circle contains and consider a transitive orientation F. from $\overline{G}$.

Claim A. Be A_1 and A_2 two different maximum cliques of G.

(a) There is an edge in F. with an endpoint in A_1 and an endpoint in A_2.

(b) All edges of $\overline{E}$, the A_1 with A_2 connect, have the same orientation in F.

Proof of Claim A. (a) If there were no such edge in F, would be like that A.$_1$ $\cup$ A.$_2$ a clique of G, a contradiction to maximality.

(b) Be away $\in$ F. and dc $\in$ F. with a, c $\in$ A.$_1$ and b, d $\in$ A.$_2$. We lead this to contradiction. Is applicablea = c or b = d, so we get from the transitivity of F. directly a contradiction. The four nodes must therefore be different in pairs. In addition, one of the edges mustad or bc in E. be included, otherwise it would be included $\overline{G}$ a sinewless one 4-Circle. Without loss of generality, we assume thatad $\in$ E. and consider the orientation in $\overline{F}$. Using the transitivity of F. follows from ad $\in$ F. direct ac $\in$ F. and from there $\in$ F. follows db $\in$ F. That is impossible because these edges are within the cliques A.$_1$respectively. A.$_2$ get lost. This concludes the proof of claim A.

Now consider the following relation on the set *C.* the maximum cliques of G: $A_1 < A_2$ if and only if an edge is in F, the A.$_1$ and A.$_2$ connects to each otherA.$_2$ is directed towards. According to claim A, this defines an oriented complete graph*C.*. We claim that (*C.*, <) is a transitive oriented clique, so*C.* linearly arranges. Let's assume thatA.$_1$ <A$_2$ and A.$_2$ <A$_3$. Then there are edgeswx, yz $\in$ F. with w $\in$ A.$_1$, x, y $\in$ A.$_2$ and z $\in$ A.$_3$. is xz $\notin$ E or wy $\notin$ E, so follows wz $\in$ F. and therefore A.$_1$ <A$_3$. So we can assume thatwy, yx and xz alone E. lie. ThereG not a sinewless 4-Circle contains follows wz $\notin$ E and the transitivity of F. implies wz $\in$ F. So A.$_1$ <A$_3$. This is proven by the assertion with the transitive-oriented clique.

We now assume that *C.* according to the above relation as A.$_1$, A$_2$,. . . , A$_m$ was arranged linearly. Suppose there are cliquesA.$_i$ <A$_j$ <A$_k$ with x $\in$ A.$_i$, x $\notin$ A$_j$ and x $\in$ A.$_k$. Therex $\notin$ A$_j$, there is a node y $\in$ A.$_j$, so that xy $\notin$ E. but A.$_i$ <A$_j$ implies xy $\in$ F, whereas A.$_j$ <A$_k$ implies that yx $\in$ F; a contradiction. That's a statement (iii) proved.

(iii) ⇒ (i) For every knot x $\in$ V denote I (x) the amount of all maximum cliques from G, the x contain. The quantitiesI (x) are intervals of the linear ordered set(*C.*, <). It remains to be shown that

$$xy \in E. \Leftrightarrow I(x) \cap I(y) \neq \varnothing \qquad (x, y \in V).$$

This is obviously the case, since two nodes are adjacent if and only if they are contained together in a maximal clique. □

From Theorem 7.1 and Proposition 1.6 we directly get the following corollary.

Corollary 7.2. *An undirected graph* G *is an interval graph if and only if it is chordal and its complement* $\overline{G}$ *is a comparability graph.*

The coloring problem and the calculation of a maximum clique, a maximum independent set or a minimum clique coverage can be solved in polynomial time with the help of the algorithms from Chapters 3 and 4. In addition, can construct a polynomial recognition algorithm by combining algorithm 1 (recognition of chordal graphs) and algorithm 7 (recognition of comparability graphs).

Statement (iii) from Theorem 7.1 has an interesting formulation using matrices. A matrix whose entries are 0 or 1 has the *Consecutive ones property for columns* if the lines can be permuted in such a way that the 1's in each column are consecutive. The clique matrix of an undirected graph $G = (V, E)$ is a matrix M, the one column for each node in V owns and one line for each maximum clique C. from G.The entry for column v and line C. is exactly then 1 if $v \in C$. The statement (iii) can then be formulated as follows.

Theorem 7.3. *An undirected graph is an interval graph if and only if its clique matrix has the consecutive ones property for columns.*

Proof. An order of the maximum cliques of G corresponds exactly to a permutation of the lines of M. The theorem follows directly from Theorem 7.1. □

This also makes it possible to obtain a more efficient recognition algorithm, namely one with a linear running time. To do this, one first calculates all maximum cliques using a perfect elimination scheme. Then the PQ trees invented by Booth and Lueker are used to obtain a representation of all clique orders in which for each node all cliques in which it is contained are consecutive. For the data structure we refer to the lecture "Algorithms for planar graphs".

The oldest characterization goes back to Lekkerkerker and Boland and dates from 1962.

Theorem 7.4 (Lekkerkerker and Boland '62, without proof). *An undirected graph* G *is an interval graph if and only if it fulfills the following conditions:*

(i) G *is chordal, and*

(ii) three nodes each of G *can be arranged so that each path from the first to the third node passes through the neighborhood of the second.*

A triple of nodes that does not satisfy property (ii) *asteroidal triple*. A graph is an interval graph if and only if it is chordal and does not contain an asteroidal triple.

7.2 Applications

Interval graphs are one of the most useful models for real-world problems. The straight line on which the intervals lie can be any one-dimensional one represent unity. The linearity often follows from physical restrictions, time restrictions or mapping by means of a cost function.

For many problems there are dedicated algorithms for interval graphs with often much better running times. In addition to the algorithms for coloring, clique coverage, independent set and maximum clique dealt with in the lecture, the existence of a Hamilton path / circle can also be efficiently decided, for example. For some problems, such as the problem of finding a cut with maximum weight, the complexity is still open.

We already got to know a first application for distributing courses to rooms in the introduction. Now let's look at two more examples.

Application 1

Be $c_1, \ldots, c_n$ Chemicals that have to be refrigerated for storage under precisely defined conditions. Any of the chemicals c_i must be at a constant Temperature between t_i and t'_i Degrees. How many refrigerators with under- different temperatures are necessary to store the chemicals?

To solve this, we consider the interval graph with nodes $c_1, \ldots, c_n$, in which we connect two nodes with each other exactly when their temperature intervals overlap. According to the Helly property (Section 3.3) it follows that if $\{c_{i_1}, \ldots, c_{i_k}\}$ a clique of G is, the intervals $\{[t_{i_j}, t'_{i_j}] \mid j = 1, \ldots, k\}$ a common cut Point t to have. A single refrigerator with temperature t is then sufficient to store them all together. A solution to the minimization problem is thus obtained by calculating a minimum clique coverage.

Application 2

Another application results from the following problem. May be X a lot of data entries and be $\mathcal{A}$. a lot of subsets of X, which are required to answer individual inquiries. Is it possible to save the data in X to be arranged in the linear memory so that for each $A \in \mathcal{A}$ the required data entries are in a contiguous memory area?

This problem can easily be written as follows. We consider one $0/1$-matrix M. with one row per data entry and one column for each request. The entry is in line i and column j exactly then 1 if the i-te data entry in the j-th request is required. Obviously, the initial question is equivalent to whether M. Has the consecutive ones property for columns. According to Theorem 7.3 this is the case if and only if the through M. defined graph is an interval graph. It should be noted that this question can of course also be solved directly using PQ trees.

7.3 Preference and indifference

May be V a lot. Suppose a decision maker meets for every pair of elements inV the decision that he strictly prefers one of them over the other or that the two elements are indifferent to him, i.e. almost completely equivalent.

With the help of these two relations (preference and indifference), two graphs can be createdH = (V, P) and G = (V, E) define as follows. For each two different elementsx, y $\in$Vis applicable

$$xy \in P. \quad \Leftrightarrow \quad \text{x is opposite y preferred,x}$$
$$xy \in E. \quad \Leftrightarrow \quad \text{and y are equivalent.}$$

By definition is H an oriented graph, G an undirected graph and (V, P + P^{-1}+ E) Completely. In fact, we can reasonably fromH Expect a lot more structure. With a rational decision maker it is to be expected thatH is at least acyclic. In fact, it is to be expected thatH is transitive; prefers someoney opposite tox and z opposite to y, so is not to be expected x and z be regarded as equivalent. In the following we therefore require thatH transitive, P. so is a partial order.

One way to quantify preferences is to use what is known as a utility function u, which each element x $\in$V a real number u (x)assigns so that x exactly then opposite y is preferred if u (x) *sufficiently larger* as u (y) is. Formally let δ>0 and u: V → R. Then is called u *Partial order utility function* for a binary relation (V, P) if the following condition is met:

$$xy \in P. \Leftrightarrow u(x) \geq u(y) + \delta \qquad (x, y \in V).$$

The following sentence answers the question which binary relations can be described by a partial order utility function.

Theorem 7.5 (Scott and Suppes '58, without evidence). *A binary relation* (V, P) *has a partial order utility function if and only if the following conditions for all* x, y, z, w $\in$V *are valid:*

(S1) P. *is irreflexive; (S2)*xy $\in$P. *and* between $\in$P.

implies xw $\in$P. *or* zy $\in$P.*(S3)* xy $\in$P. *and* Y Z $\in$P.

implies xw $\in$P. *or* wz $\in$P..

A relation that fulfills the three conditions (S1), (S2) and (S3) is also called *Semi-order*. Now we consider the indifference relationG = (V, E) a partial order (V, P)which is a partial order utility function u allows. Two different knotsx andy are in G neighboring if and only if |u (x) - u (y) | <δ.With that you can G as

Represent interval graph in which each node x is represented by an open interval of length δ, the center of which u (x) is. This means that indifference graphs are unit interval graphs, that is to say interval graphs that have a representation in which all intervals have the same length, without restriction on length1.

Theorem 7.6 (Roberts '69). *May be* G = (V, E) *an undirected graph. The following statements are equivalent.*

(i) There is a real-valued function u: V → R.*so that for two different nodes each* x, y ∈ V *is applicable*

$$xy \in E. \Leftrightarrow |u(x) - u(y)| < 1.$$

(ii) There is a semi-order (V, P)*, so that* E = P + $\overline{P^{-1}}$.

(iii) $\overline{G}$ *is a comparability graph and any transitive orientation of* $\overline{G}$ *is a semi-order.*

(iv) G *is an interval graph that is not an induced copy of* $K_{1.3}$ *contains.*

(v) G *is a real interval graph.*

(vi) G *is a unit interval graph.*

Proof. (i) ⇒ (vi): May be u the function from (i). We associate with each knotx the interval I_x = (u (x) - $\frac{1}{2}$, u (x) + $\frac{1}{2}$). Obviously,$I_x \cap I_y$ *6th*= ∅ if and only if |u (x) - u (y) | <1.So is $\{I_x\}_{x \in V}$ an interval representation of G with unit intervals.

(vi) ⇒ (v): Since no unit interval can genuinely contain another unit interval, an interval representation with unit intervals is always genuine.

(v) ⇒ (iv): Accepted G would contain one to $K_{1.3}$ isomorphic subgraphs $G_{Y\,Z_1,\, e.g._2,\, e.g._3}$. Where {z1, e.g.2, e.g.3} an independent set and y is to each of the z_i adjacent. isI_{z_j} the interval I_{z_1}, I_{z_2}, I_{z_3} the lies completely between the other two, it follows that from that I_y the interval I_{z_j} must contain completely. That's a contradiction, so can G no induced subgraph is isomorphic to it $K_{1.3}$ contain.

(iv) ⇒ (iii): There G is an interval graph is $\overline{G}$ = (V, $\overline{E}$) a comparability graph. May beF. a transitive orientation of $\overline{G}$. Obvious is F. irreflexive and thus fulfills property (S1). Suppose it wasxy ∈ F. and between ∈ F. There G does not contain an induced circle of length 4 by Theorem 7.1, there is another edge in $\overline{E}$. on the set of nodes {x, y, z, w}. By means of transitivity it then follows that xw ∈ F. or zy ∈ F; F.thus satisfies property (S2). We now show property (S3). Bexy ∈ F. and Y Z ∈ F,but xw ∉ F and wz ∉ F. According to the transitivity of F. follows that wx ∉ F and between ∉ F and that wy ∉ F and yw ∉ F, but xz ∈ F. So is $G_{x,\,y,\,z,\,w}$ isomorphic to $K_{1.3}$. A Contradiction.

(iii) ⇒ (ii): Follows directly.

(ii) ⇒ (i): Is (V, P) a semi-order, then there is a real-valued function u': $V \to \mathbb{R}$.and a number $\delta > 0$, so that $xy \in P$. exactly when $u'(x) - u'(y) \geq \delta$. We define$u(x) = u'(x) / \delta$. There $P + P^{-1} = \overline{E}$ is applicable

$$xy \in E. \Leftrightarrow |u(x) - u(y)| < 1.$$

□