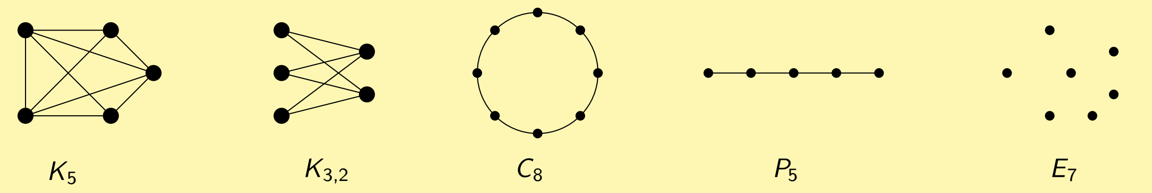
Algorithmic Graph Theory

Introduction

A graph is always simple, undirected, has no loops and has a finite vertex set.
For $n \in \mathbb{N}_0$, $[n] := \{1, 2, \dots, n\}$
For sets A, B, C, we write $A + B = C$ iff. $A \cup B = C$ and $A \cap B = \emptyset$
We call $V_1 + \dots + V_t = V$ a partition of V in t parts.

Example / Important Graphs



- Clique Number $\omega(G)$: Maximal clique in G
- Independence Number $\alpha(G)$:Maximal independent Set of G
- Chromatic Number $\chi(G)$: Minimal partition into independent sets of G
- Clique Cover Number $\kappa(G)$: Minimal partition of $V(G)$ into cliques

Example

	K_n	$K_{n,m}$	C_n	P_n	E_n
$\omega(G)$	n	2	2, if $n \geq 4$ 3, if $n = 3$	2, if $n \geq 2$	1
$\alpha(G)$	1	$\max(n, m)$	$\lfloor \frac{n}{2} \rfloor$	$\lceil \frac{n}{2} \rceil$	n
$\chi(G)$	n	2	2, if n is even 3, if n is odd	2, if $n \geq 2$	1
$\kappa(G)$	1	$\max(n, m)$	$\lceil \frac{n}{2} \rceil$, if $n \geq 4$	$\lceil \frac{n}{2} \rceil$	n

Observation:
$\omega(G) \approx \chi(G)$ and
$\alpha(G) \approx \kappa(G)$
in the examples

Lemma K1:
For all graphs G:

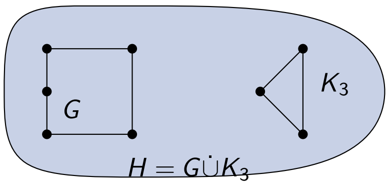
- $\chi(G) \geq \omega(G)$
- $\kappa(G) \geq \alpha(G)$

Proof Sketch: Each vertex in a clique needs different color

Goal of the Lecture: Examine graphs for which Equality holds.

Boring answer: Consider any graph G with chromatic number t, then $H := G \dot\cup K_t$ has the property

$$\chi(H) = \max(\chi(G), \chi(K_t)) = t = \omega(K_t) = \omega(H).$$



$\chi(H) = 3$

$\omega(H) = 3$

To 'exclude' such boring examples, we consider the following definitions:

Let G be a graph.

- We say G has property P1, iff. for all $A \subseteq V(G)$, $\chi(G_A) = \omega(G_A)$.
- We say G has property P2, iff. for all $A \subseteq V(G)$, $\kappa(G_A) = \alpha(G_A)$.
- We call G perfect, if it satisfies P1 and P2.

Note: G is perfect, iff. the complement $\bar{G}$ of G is perfect.

Examples:

- K_n is perfect, as each induced subgraph of K_n is also complete, which satsifies the requierment (see example above).
- E_n is perfect.
- $K_{n,m}$ is perfect.
- P_n is perfect, as each induced subgraph is a disjoint union of paths (for which the requierment holds).
- C_n is perfect iff. n is even.
- Any graph with less than 5 vertices is perfect.

Weak Perfect Graph Theorem (WPGT)

The Weak Perfect Graph Theorem (WPGT):
An arbitrary graph G satisfies property P1 iff. it satisfies property P2.

The next pages contain the proof of WPGT.

Definition Property P3:
We say that a graph G satisfies property P3, if for all $A \subseteq V(G)$,

$$\omega(G_A)\alpha(G_A) \geq |A|.$$

Example:
C_5 has **not** P3, as $\alpha(C_5)\omega(C_5) = 2{\cdot}2 < 5 = |C_5|$

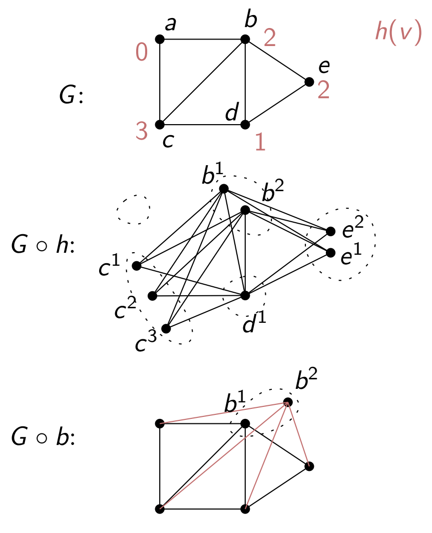
We will show: P1 $\iff$ P3 $\iff$ P2.

Definition Vertex Repitition Graph:
Given a graph G and a mapping $h : V(G) \to \mathbb{N}^V$, we define $G \circ h$ by

$$V(G \circ h) = \bigcup_{v \in V(G)} \{v^1, \ldots, v^{h(v)}\}$$

$$E(G \circ h) = \{u^i v^j \mid uv \in E(G), i \in [h(u)], j \in [h(v)]\}$$

Furthermore, for any vertex $v \in V(G)$, let $G \circ v := G \circ v$, where $h(v) = 2$ and $h(u) = 1$ for all $u \neq v$.



We note that for every h and G, the graph $G \circ v$ can be obtained from G by repeatetly applying $G \circ v$ and $G - h$.

Lemma 2.6:
For any graph G and mapping h,

- G has P1 $\iff$ $H := G \circ h$ has P1 and
- G has P2 $\iff$ $H := G \circ h$ has P2.

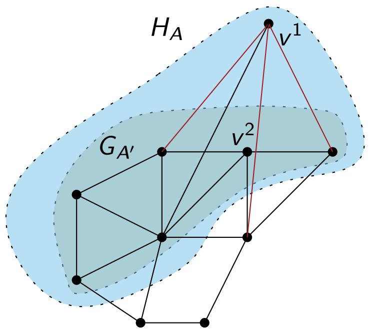
Proof of Part 1:

- w.l.o.g. $H = G \circ v$ or $H = G - v$ for a vertex $v \in V(G)$ (see observation above).
- If $H = G - v$, then H is an induced subgraph of G. An induced subgraph of a graph with P1 has also property P1. (done)
- Let $H = G \circ v$ for a vertex $v \in V(G)$.
- Note that $H - v^1 \cong G \cong H - v^2$.
- Take any $A \subseteq V(H)$. We need to show that $\chi(H_A) = \omega(H_A)$.
- If $v^1 \notin A$ or $v^2 \notin A$, then $A \subseteq V(G)$ (up to isomorphism). As G has P1, we have $\chi(H_A) = \chi(G_A) = \omega(G_A) = \chi(H_A)$. (done)
- Thus, we consider the remaining case $\{v^1, v^2\} \subseteq A$.
- Let $A' := A - v^1$. We note:

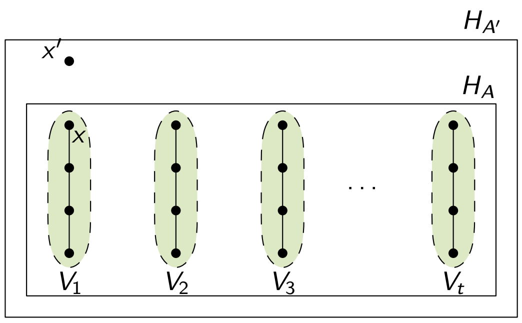
$$\chi(H_A) \leq \chi(G_{A'}) \underset{\text{P1 of } G}{=} \omega(G_{A'}) \underset{G_{A'} \subseteq H_A}{\leq} \omega(H_A) \underset{\text{Lemma K1}}{\leq} \chi(H_A)$$

For any coloring of $G_{A'}$, we obtain a coloring of H_A with the same number of colors, by copying all colors and assigning v^1 the same color as v^2.

- Thus, $\chi(H_A) = \omega(H_A)$ (done)

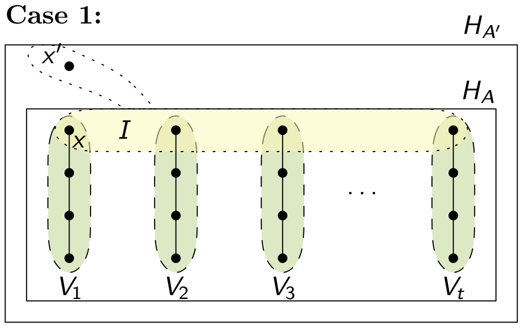


Proof of Part 2



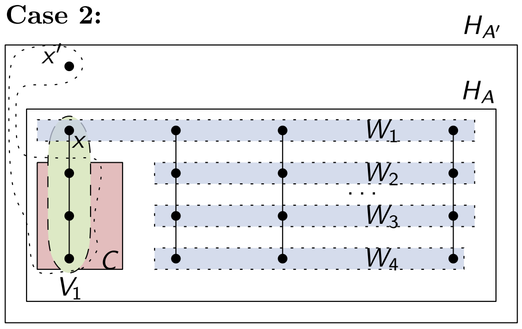
- Again, w.l.o.g. $H = G \circ x$ or $H = G - x$, where the second case is again trivial.
- Let x and x' be the copies of x in H.
- Let G satisfy property P2. We will now show that H also satisfies P2.
- Let $A' \subseteq V(H)$. The only non-trivial case is, when $\{x, x'\} \subseteq A'$.
- We define $A := A' - x'$ and note that $A \subseteq V(G)$.
- As G satisfies P2, we have $\kappa(G_A) = \alpha(G_A)$.
- Thus, we find a clique cover $V_1 + \dots + V_t$ of $G_A = H_A$ with $t = \alpha(H_A)$.
- We distinguish two cases:

Case 1:



- Case 1: There exists an i-set I of H_A such that $|I| = t, x \in I$.
  - Now, $I + x'$ is an i-set of $H_{A'}$ of size $\alpha(H_A) + 1 = t + 1$. Thus, $\alpha(H_{A'}) \geq t + 1$.
  - Additionally, $V_1 + \dots + V_t + \{x'\}$ is a clique cover of $H_{A'}$. Thus, $\kappa(H_{A'}) \leq t + 1$.
  - Hence, $\kappa(H_{A'}) \leq t + 1 \leq \alpha(H_{A'}) \leq \kappa(H_{A'})$ (done)

Case 2:



- Case 2: For all i-set I of H_A with $|I| = t$, $x \notin I$.
  - W.l.o.g. let $x \in V_1$.
  - Let $C := V_1 - x$ and consider H_{A-C}.
  - Now, $\alpha(H_{A-C}) \leq t - 1$ (any maximal clique of H_A contained a vertex in V_1, thus C)
  - P2 for G yields a clique cover $W_1 + \dots + W_s$ of H_{A-C} with $s = \alpha(H_{A-C}) \leq t - 1$ cliques.
  - Note that $C + x = V_1$ is a clique and thus $C + x'$ is also a clique.
  - $W_1 + \dots + W_s + (C + x')$ is now a clique cover of $H_{A'}$.
  - Thus, $\kappa(H_{A'}) \leq s + 1 \leq t - 1 + 1 = t$
  - In total:

$$\kappa(H_{A'}) \underset{\text{see above}}{\leq} t \underset{\text{Definition } t}{=} \alpha(H_A) \underset{\text{subgraph}}{\leq} \alpha(H_{A'}) \underset{\text{for all graphs}}{\leq} \kappa(H_{A'})$$

(done)

For the proof of the WPGT, we only need one additional Lemma:

Lemma 2.7
If G is a graph with property P3 such that any *proper* induced subgraph of G has property P2, then $H := G \circ h$ also has property P3.

Recall:

Definition Property P3:
We say that a graph G satisfies property P3, if for all $A \subseteq V(G)$,

$$\omega(G_A)\alpha(G_A) \geq |A|.$$

Proof of Lemma 2.7

- For sake of contradiction, assume that P3 does not hold for H.
- W.l.o.g. let H be the smallest such counterexample, thus
  - $\underbrace{\omega(H)}_{w:=}\underbrace{\alpha(H)}_{a:=} < |V(H)|$
  - $\omega(H_A)\alpha(H_A) \geq |A|$ for all $A \subsetneq V(H)$,
- If for all $v \in V(G)$, $h(v) \leq 1$, then the case is trivial.
- Thus, let there be $s \in V(G)$ such that $h(s) \geq 2$. Let $S = \{s_1, \dots, s_z\}$ be the $z := h(s) \geq 2$ copies of s in H.
- We consider the graph $H - s_z$, which has property P3 (as H is a minimal counterexample), thus

subgraph of H

$$|V(H)| - 1 = |V(H - s_z)| \leq \omega(H - s_z)\alpha(H - s_z) \leq wa \leq |V(H)| - 1.$$

- Thus, $wa = |V(H)| - 1$, $\omega(H - s_z) = w$ and $\alpha(H - s_z) = a$.
- Now, $\alpha(H - S) = a$, because if $\alpha(H - S) < a = \alpha(H)$, then every biggest i-set I of H must contain a vertex $s_i \in S$, but then I must contain s_z, as otherwise $I + s_z$ would be a bigger i-set. However, then $\alpha(H - s_z) \leq |I - s_z| = a - 1$ (Contradiction to the last bullet point).
- We now consider $G - s$. We note that $G - s$ has property P2 by requierment.
- As $H - S$ is obtained by vertex multiplication of $G - s$, we note by Lemma 2.6 that $H - S$ has property P2 too.
- Thus, we can find a clique cover $V_1 + \dots + V_a$ of $H - S$.

$$(\kappa(H - S) = \alpha(H - S) = a)$$

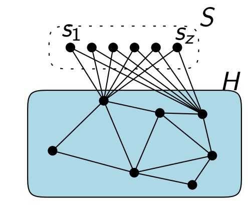
- We note:

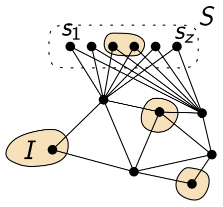
$$wa \geq |V(H - S)| = |V(H)| - |S| = wa + 1 - z = wa - (z - 1)$$

- Hence, at most $z - 1$ of the cliques V_i can be non-maximal i.e. have size $< w$. W.l.og. let $|V_i| = w$ for all $i \in [a - (z - 1)]$.
- Let X be the union of all V_i with $i \in [a - (z - 1)]$ and $\{s_1\}$.
- Note that $|X| = (a - (z - 1))w + 1$ and $\omega(H_X) = w$.
- As $H_X \subseteq H$ and H is a minimal counterexample, we know that H_X satisfies property P3, thus:

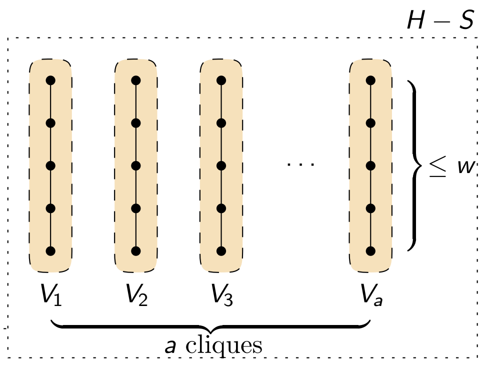
$$\alpha(H_X) \geq \frac{|X|}{\omega(H_x)} = \frac{(a - (z - 1))w + 1}{w} = a - (z - 1) + \frac{1}{w}.$$

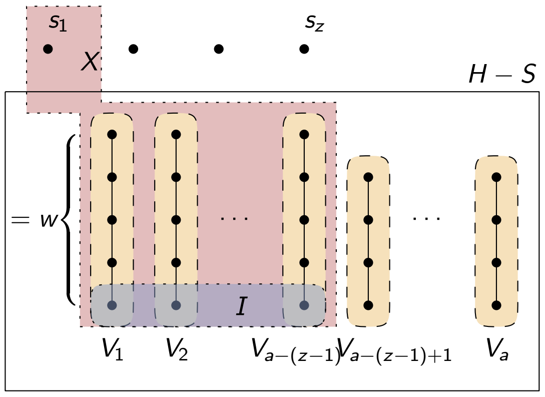
- As $\alpha(H_X) \in \mathbb{Z}$ and $1/w < 1$, we obtain $\alpha(H_X) \geq a - (z - 1) + 1$.
- Thus, there exists an i-set I of H_X of size $a - (z - 1) + 1$. Note that $s_1 \in I$, as $\alpha(H_X - s_1) \leq \kappa(H_X - s_1) = a - (z - 1)$.
- Now, $I + \{s_2, \dots, s_z\}$ is an i-set of H of size $a - (z - 1) + 1 + (z - 1) = a + 1$. (Contradiction)





Any i-set I of H that contains at least one, but not all, vertices in S cannot be maximal, as in that case adding the remaining vertices of S to I would keep it independent.





Proof of the WPGT:
We prove by induction over the number of vertices n of a graph G, that

$$G \text{ has P1} \iff G \text{ has P2} \iff G \text{ has P3}. \tag{IH}$$

The base case is trivial. We assume that for a graph G the claim (IH) already holds for all graphs of smaller size and show that (IH) is also satisfied by G. We show the induction step in multiple parts:

Claim 1: G has P1 $\implies$ G has P3

- Assume G has P1.
- Let $A \subseteq V(G)$. We want to show that $\alpha(G_A)\omega(G_A) \geq |A|$.
- This is trivial, if $A \subsetneq V(G)$, by the induction hypothesis (IH). Thus, we only consider the case $A = V(G)$ i.e. we want to prove $\alpha(G)\omega(G) \geq |V(G)|$.
- As G has P1, there exists a coloring $V_1 + \cdots + V_t$ of G with $t = \chi(G) = \omega(G)$ colors. Note that each color class V_i is an i-set, and thus, of size $\leq \alpha(G)$.
- We conclude $|V(G)| = \sum_{i=1}^{\omega(G)} |V_i| \leq \sum_{i=1}^{\omega(G)} \alpha(G) = \omega(G)\alpha(G)$.

Claim 2: G has P3 $\implies$ G has P1

- Assume G has P3. Similar as in Claim 1, we only need to show that $\omega(G) = \chi(G)$ (i.e. we need to only consider the case $A = V(G)$).
- Let $\mathcal{C}_+$ be the set of all cliques of G of size $\omega(G)$. We distinguish two cases:
- Case 1: There exists an i-set I of G such that $C \cap I \neq \emptyset$ for all $C \in \mathcal{C}_+$.
  - Consider the graph $G - I$ and note that $\omega(G - I) = \omega(G) - 1$.
  - Additionally note that by (IH), $G - I$ has P1, thus there exists a coloring $V_1 + \cdots + V_t$ of $G - I$ with $t = \chi(G - I) = \omega(G - I) = \omega(G) - 1$.
  - Now, $V_1 + \cdots + V_t + I$ is a coloring of G with $\omega(G)$ colors.
  - Together with $\chi(G) \geq \omega(G)$ (for all graphs), this finishes this case.
- Case 2: For any i-set I of G there exists a maximal clique $C(I) \in \mathcal{C}_+$ such that $I \cap C = \emptyset$.
  - Let $\{v_1, \dots, v_n\} := V(G)$. We define $h_i = |\{I \subseteq V(G) \text{ i-set} \mid v_i \in C(I)\}|$
  - Now, consider $H := G \circ h$ (where $h = (h_1, \dots, h_n)$). Note that $H \neq \emptyset$
  - We requiered P3 for G and thus P3 for any subgraph of G. By (IH), we thus have (P2) for any proper subgraph of G. With Lemma 2.7 it follows (P3) for H.
  - If we set $X = V(H)$, we obtain $\omega(H)\alpha(H) \geq |X|$.
  - By definition of H and noting that each i-set of G increments exactly $\omega(G)$ many of the counter h_i, we have $|X| = \sum_{i \in [n]} h_i = \omega(G) \cdot |Y|$, where Y is the set of i-sets of G.
  - Also note $\omega(H) \leq \omega(G)$ (vertex multiplication does not yield bigger cliques).
  - Note

Note that any maximal i-set of H must be based on a (not necessaraly maximal) i-set of G, where we include all copies of vertices in the original i-set.

$$\alpha(H) = \max\left\{\sum_{v_i \in I} h_i \mid I \subseteq V(G) \text{ is i-set}\right\}$$

Note that each increment in $\sum_{v_i \in I} h_i$ corresponds to a vector $v_i \in I$ beeing contained in $C(I')$ for some i-set I'. Now, $|C(I') \cap I| = 1$ iff. there exists an $v_i \in I$ with $v_i \in C(I')$.

$$= \max\left\{\sum_{I' \subseteq V(G) \text{ i-set}} |C(I') \cap I| \mid I \subseteq V(G) \text{ is i-set}\right\} \leq |Y| - 1$$

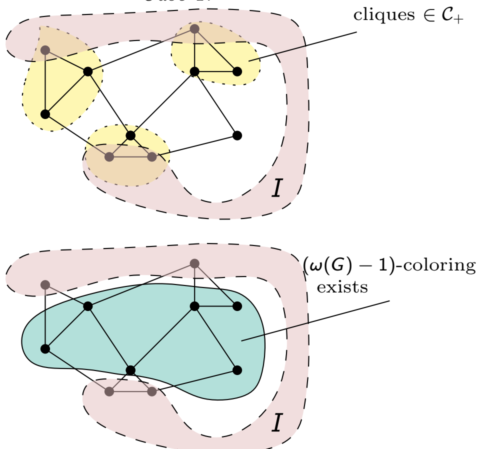
each I' is in Y. But $C(I) \cap I = 0$.

  - We conclude

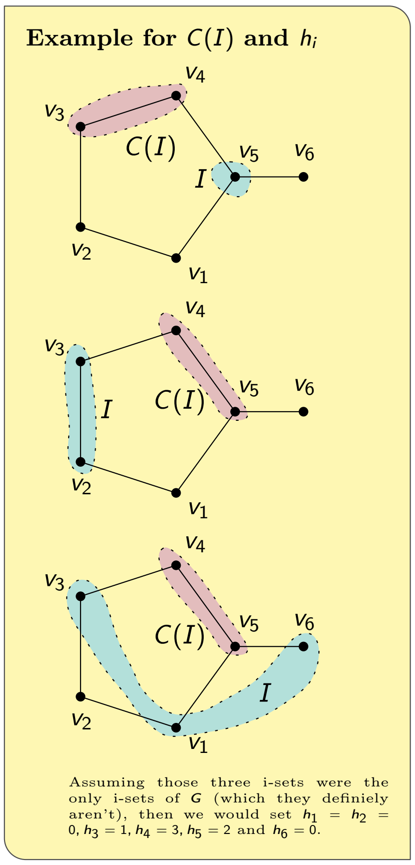
$$\omega(G)(|Y| - 1) \geq \omega(H)\alpha(H) \geq |X| = \omega(G)|Y|$$

  - This is a contradiction, thus Case 2 can never occure.

Case 1:



Case 2:



Assuming those three i-sets were the only i-sets of G (which they definiely aren't), then we would set $h_1 = h_2 = 0$, $h_3 = 1$, $h_4 = 3$, $h_5 = 2$ and $h_6 = 0$.

Claim 1: G has P2 $\iff$ G has P3

- Note:

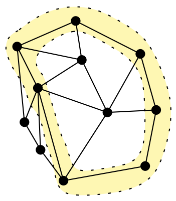
Note the correspondence between ω, χ and α, κ when considering graph complements.

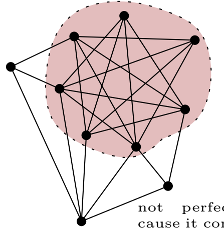
$$G \text{ has (P2)} \iff \overline{G} \text{ has (P1)} \overset{\text{by Claim 1 and 2}}{\iff} \overline{G} \text{ has (P3)} \iff G \text{ has (P3)}$$

This finishes the proof of WPGT.

Chordal Graphs

For a graph G, $A \subseteq V(G)$ and $t \geq 4$, we call A (or G_A) a ***t*-hole** of G, if $G_A \cong C_t$. Similarly, we call A (or G_A) a t-antihole, if $G_A \cong \overline{C_5}$ (i.e. $\overline{G}$ contains a t-hole). We note that a graph containing an odd hole or odd antihole (i.e. $t \geq 4$, t odd) cannot be perfect. The Strong Perfect Graph Theorem states that the converse is also true:


not perfect, because it contains 7-hole.


not perfect, because it contains 7-antihole.

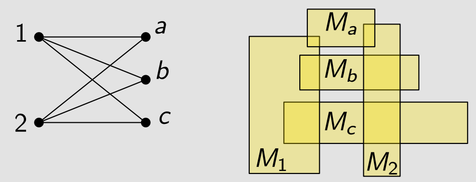
Strong Perfect Graph Theorem (SPGT)
A graph G is perfect $\iff$ G contains no odd hole and no odd antihole.

The proof is complicated and will be skipped. (TBD: is this true?)

We now consider some graph classes that are perfect.

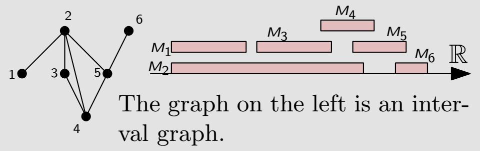
Definition Intersection Graphs:
Give an universe $\mathcal{U}$ containing sets. , we call a graph G a $\mathcal{U}$-intersection graph, if there exists an assignment M_v for each vertex $v \in V(G)$ such that for all $u, v \in V(G)$:

$$uv \in E(G) \iff M_u \cap M_v \neq \emptyset$$



$K_{2,3}$ is a $\mathcal{R}$-intersection graph, if we define $\mathcal{R} \subseteq \mathbb{R}^2$ as the set of all rectangular boxes.

Definition Interval Graph:
G is called an interval graph, if G is a $\mathcal{I}$-intersection graph with $\mathcal{I} = \{[a, b] \mid a, b \in \mathbb{R}\}$



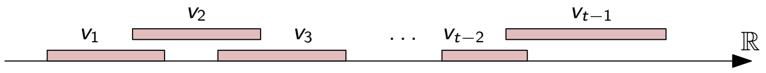
The graph on the left is an interval graph.

We will show that all interval graphs are perfect.

Lemma
Let G be an interval graph, then G contains no t-hole ($t \geq 4$).

Proof (by Picture):

- Assume G did contain a t-hole $\{v_1, \dots, v_t\}$ with $t \geq 4$.
- Consider how the interval of the vertices $v_1, \dots, v_{t-1}$ might be placed and note that it must look similar to:



- Note that any placement of v_t must intersect with the left most and the right most interval, but not with any interval inbetween, which is impossible.

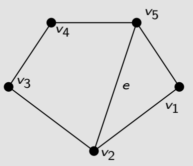
(done)

This Lemma alone does not show that an interval graph is perfect, as we must show that no odd antiholes exist too. (After that we might apply the SPGT). To show that, we observe that interval graphs are contained in a more general graph class called chordal graphs:

Definition Chordal Graph:
A graph G is called chordal, if G contains no t-hole with $t \geq 4$ (note that we do not requiere oddness of the hole).

Equivalent:
A graph G is chordal, iff any cycle of length ≥ 4 of G has a chord.

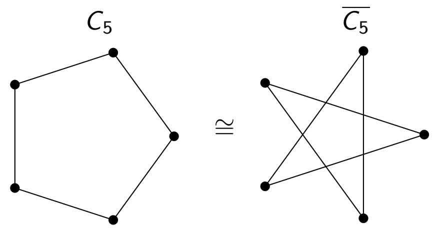
The 5-cycle $(v_1, \dots, v_5)$ in this graph has a chord (e). However, there is a 4-cycle $(v_2, \dots, v_5)$ that has no chord.
$\implies$ not chordal



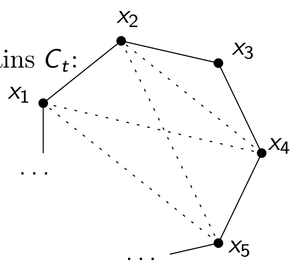
Lemma: Any chordal graph G is perfect.

Proof (using the SPGT):

- We apply SPGT, thus we want to show that G contains no odd holes or antiholes.
- (Odd) holes cannot occure in G by definition of chordal graphs.
- Assume G contains an antihole $x_1, x_2, \dots, x_t$ with odd t.
- If $t = 5$, then note $\overline{C_5} \cong C_5$, thus G also contains a t-hole (Contradiction).
- If $t > 5$, then x_1, x_4, x_2, x_5 is a 4-hole in G (Contradiction).



$\overline{G}$ contains C_t:



For the proof of "G is chordal $\implies$ G is perfect", we used the SPGT, which we have not proven. As this is unsatisfying, we will prove this statement again, without use of SPGT. For that reason we first explore properties of chordal graphs.

What we will observe is that chordal graphs behave (in a sense) similar to trees. ("You can think about chordal graphs as thick trees.")
Two usefull properties of trees are:

- Trees can be easily separated into smaller graphs (remove root)
- Trees have leaves.

We start by generalizing the second property;

Definition Simphicial
We call a vertex $v \in V(G)$ simphicial, if $Adv(v) := N(v) := \{u \in V(G) \mid uv \in E(G)\}$ is a clique.

Lemma 3.6:
Any chordal graph has a simphicial vertex.

It will take some work (and axillary lemmas) to prove Lemma 3.6. Before we do so, we show, why Lemma 3.6 is relevant.

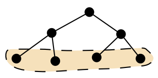
Lemma A
If v is simphicial in G and $G - v$ is perfect, then G is perfect.

Proof of Lemma A:

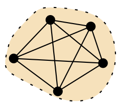
- By the WPGT, we only need to verify (P1) for G.
- Consider any $A \subseteq V_G$. We will show that $\omega(G_A) = \chi(G_A)$.
- If $v \notin A$, then $A \subseteq V(G - v)$, and we are done (as $G - v$ is perfect).
- We now assume $v \in A$. Let $A' := A - v \subseteq V(G - v)$.
- Now, $\chi(G_{A'}) = \omega(G_{A'})$ and there exists a coloring $V_1 + \cdots + V_t$ of $G_{A'}$ with $t = \omega(G_{A'})$.
- Case 1: $|Adj(v) \cap A'| < t$.
  - Color G_A in t colors, by giving v a color, non of it's $< t$ neighbors has.
  - Now,
  $$\chi(G_A) = t = \chi(G_{A'}) = \omega(G_{A'}) \leq \omega(G_A)$$
- Case 2: $|Adj(v) \cap A'| \geq t$.
  - As $Adj(v) \cap A'$ is a clique (as v is simphicial), $|Adj(v) \cap A'| \leq \omega(G_{A'}) = t$. Hence $t = |Adj(v) \cap A'|$.
  - Obviously G_A can be colored with $t + 1$ colors.
  - $(Adj(v) \cap A') \cup \{v\}$ is a clique of size $t + 1$ of G_A. Hence,
  $$\omega(G_A) \geq t + 1 = \omega(G_{A'}) + 1 = \chi(G_{A'}) + 1 \geq \chi(G_A) \geq \omega(G_A)$$
  (done)

Examples for Chordal Graphs

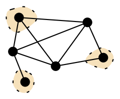
- complete and empty graphs
- paths
- trees and forests
- interval graphs



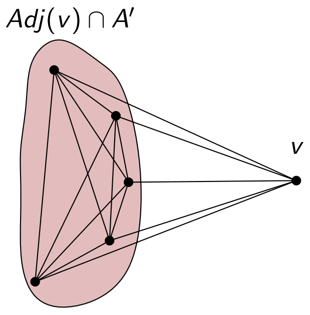
Trees:
Leaves are simphicial



Complete Graphs:
All vertices are simphicial



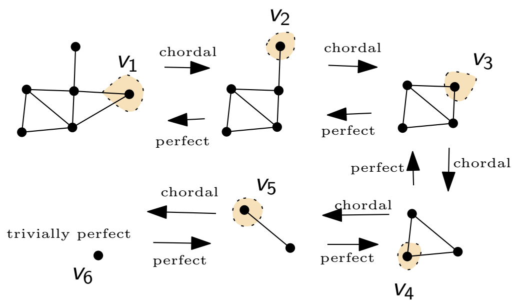
Example:
3 vertices are simphicial



Lemma 3.6 (which we still have to prove) and Lemma S1 together imply that any chordal graph is perfect.

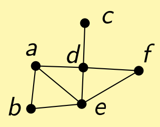
Proof:

- We show this by induction over $n = |V(G)|$. The base case is trivial. Induction Step:
  - Assume all chordal graphs with less then n vertices are perfect.
  - Let G be chordal with $|V(G)| = n$.
  - By Lemma 3.6, G has a simphicial vertex v.
  - Now, $H := G - v$ is still chordal, as an induced subgraph of G and thus perfect by induction hypothesis
  - By Lemma S1, G is perfect (done)

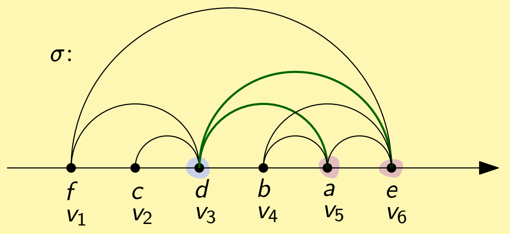


The idea used in this proof does not only work on chordal graphs, but can be generalized:

Definition PES:
For a graph G with n vertices, we call a vertex ordering $\sigma = (v_1, \dots, v_n)$ a *perfect elimination scheme (PES)*, if for all i, v_i is simphicial in $G_{\{v_i,\dots,v_n\}}$.



The following vertex ordering of the graph above, is a PES, as all "right neighbors" of a vertex (highlighted for v_3) form a clique.



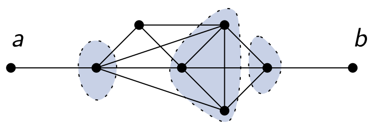
We say things like "The vertex c ordered left of a" or "The vertex b is directly after d."

With a very similar argument as above (lightblue box), one can show that any graph G that has a PES is perfect.

Back to analyzing chordal graphs: We need to introduce the concept of a separator:

Definiton Separator:
For a graph G, we call a set $S \subseteq V(G)$ a separator, if $G - S$ is disconnected. For any non-adjacent vertices $u, v \in V(G)$, we call S an $a.b$-separator, if a and b are in different connected components of $G - S$.

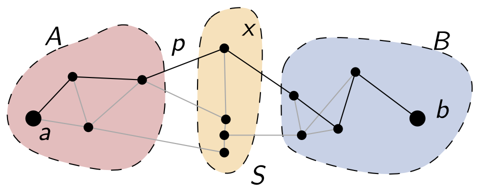
Lemma 3.4
If G is a chordal graph, $a, b \in V(G)$ are non-adjacent vertices and S is an inclusion-minimal $a.b$-separator, then S is a clique.

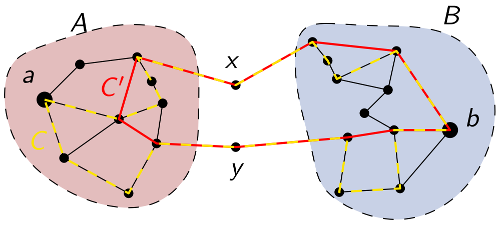


All 3 sets (blue) are *inclusion-minimal* $a.b$-separators. Note that there are no other inclusion-minimal $a.b$-separators in this graph.

Proof:

- If $|S| = 1$, then S is trivially a clique. Thus, we assume $|S| \geq 2$.
- Take any $x, y \in S, x \neq y$. We need to show that $xy \in E(G)$.
- Let A (respecively B) be the connected component of x (respectively y) in $G - S$.
- As S is inclusion-minimal, $S - x$ is no $a.b$-separator.
- Thus, there exists a (simple) path p from a to b that uses no vertex from $S - x$. This can no longer be a path in $G - S$, thus p must use x.
- Note that one of the neighbors of x in p must be in A and the other in B.
- Analogously, we can conclude that y is also directly adjacent to a vertex in A and a vertex in B.
- Now, we can find a cycle C in G that passes through x, then a, then y, then b and then x again (in this order, but with other vertices allowed inbetween).
- We can shorten C to get a cycle C' of the form $(x, a_1, \dots, a_t, y, b_1, \dots, b_k, x)$ with $a_i \in A$ and $b_i \in I$. Let C' be the shortest cycle of this form.
- Now, $|C'| \geq 4$ and G is chordal, thus C' has a chord e. We distinguish:





Case 1: $e = a_ia_j$ or $e = b_ib_j$:
Then, we can shorten C' by taking e as a shortcut. (Contradiction, as C' is shortest cycle)

Case 2: $e = a_ib_j$ or $e = a_jb_i$:
Impossible, as then S would be no $a.b$-separator.

Case 3: $e = a_ix$ or $e = a_iy$ or $e = b_ix$ or $e = b_iy$:
Again, as in case 1, we could use e as a shortcut (Contradiction)

Case 4: $e = xy$
This is what we wanted to show.

(done)

We can now prove Lemma 3.6. Recall:

Lemma 3.6:
Any chordal graph has a simphicial vertex.

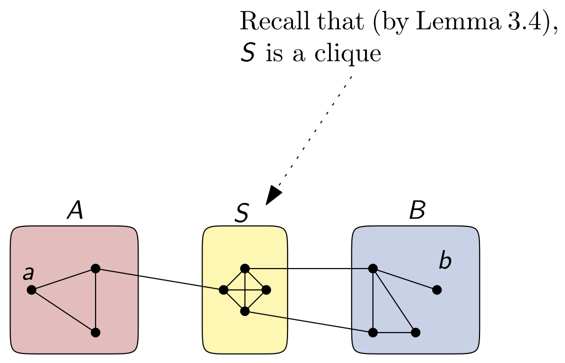
We will prove the following stronger statement

Stronger Version of Lemma 3.6
For any chordal graph G,

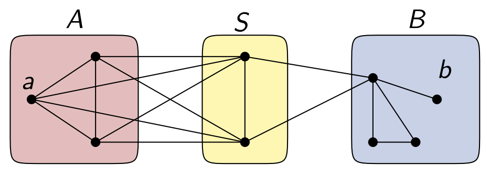
- G has a simphicial vertex
- If $G \neq K_n$, then G has two *non-adjacent* simphicial vertices.

Proof (by induction over the number of vertices $n = |V(G)|$):

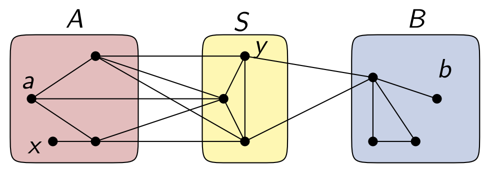
- For $n = 1$, $G = K_1$ and the unique vertex of G is simphicial.
- Now, assume the claim holds for all chordal graphs with less then n vertices, for some n.
- Let G be an arbitrary graph with n vertices.
- **Case 1:** $G = K_n$
  - Choose any vertex of G. It is simphicial.
- **Case 2:** $G \neq K_n$
  - Then there exist non-adjacent $a, b \in V(G)$.
  - Let S be any *inclusion-minimal* $a.b$-separator of G. Let A (respectively B) be the connected component of a (respectively b) in $G - S$.
  - **Case 2a:** $G_{A+S} = K_n$
    - Note that all neighbors of a (in G) are in A or S.
    - Thus, the neighbors of a are a clique. ($\implies$ a is simphicial in G)
  - **Case 2b:** $G_{A+S} = K_n$
    - By the induction hypothesis, there exist two *non-adjacent* vertices $x, y \in A + S$ that are simphicial in G_{A+S}.
    - By Lemma 3.4, S is a clique. Thus, not both x and y can be in S.
    - W.l.og. $x \in A$. The neighbors of x in G are exactly the neighbors of x in G_{A+S}, thus x is also simphicial in G.
  - In both cases we found a vertex $u \in A$ that is simphicial in G. Similarly, we find a vertex $v \in B$ that is also simphicial in G.
  - u and v are obviously non-adajcent. (done)



Case 2a:



Case 2b:



In the previous Lemmas, we have seen (more or less) the following properties of chordal graphs:

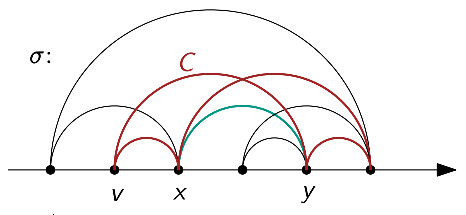
$$\begin{tikzcd}
\text{G is chordal} \arrow[r, Leftrightarrow, "\text{Def.}"] & \text{Any cycle of length ≥ 4 of G has a chord.} \arrow[r, Rightarrow, "\text{Lemma 3.4}"] & \text{Every inclusion-minimal separator of G is a clique} \arrow[d, Rightarrow, "\text{Lemma 3.6}"] \\
& \text{G has a PES.} \arrow[ul, dashed, "\text{(Lemma X)}"] & \text{Every induced subgraph of G has a simphicial vertex.} \arrow[l, Rightarrow, "\approx \text{Lemma A}"']
\end{tikzcd}$$

We now show that all these statements are equivalent. For that reason, we show:

Lemma X:
If G has a PES, then G is chordal.

Proof of Lemma X:

- Let σ be a PES of G and C a cycle of length ≥ 4. We show that C has a chord.
- Let v be the leftmost vertex in C (according to σ) and let x, y be the neighbors of v in C.
- As x and y are right neighbors of v, and σ is a PES, x and y must be adjacent.



(done)

Recognition of Chordal Graphs

The Chordal Graph Recognition Problem (CGRP)
Given a graph G, decide whether G is chordal or not.

Trivial Algorithm: Go over all vertices ($O(n)$) and check whether they are simphicial (worstcase $O(n^2)$ checks). If no simphicial vertex v is found, G is non-chordal. Otherwise, repeat the procedure on $G - v$ ($O(n)$ repetitions).
Runtime: $O(n^4)$ or more fine-grained: $O(n^2(n+m))$.

We will show that CGRP can be solved in linear time. With this goal in mind, we first consider the algorithm LexBFS. It can be summarized as follows:

- Do BFS on G and number / label the vertices in the order of exploration from n to 1.
- At any step, use the following principle as a tie-break: If the label of a vertex u comes lexicographically after the lable of a vertex v, then u must be explored before v.

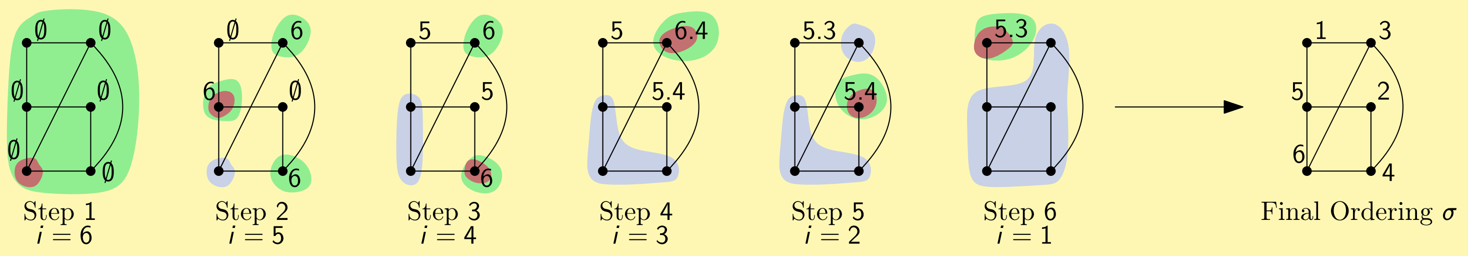
Input : undirected graph $G = (V, E)$.
Output : vertex ordering σ.

```
assign each vertex label $\emptyset$;
for $i \leftarrow n$ to 1 do
    choose a vertex $v$
        with no assigned number in $\sigma$
            with lexicographically largest label;
    $\sigma(i) \leftarrow v$;
    for every vertex $w \in \mathrm{Adj}(v)$
        with no assign number in $\sigma$
        append $i$ to label$(w)$;
    end for
end for
```

Algorithm 1 : LexBFS

Example for Lexicographic Order:
$\emptyset < 5 < 5.3 < 5.4 < 6.1 < 6.1.2 < 7$

Example Execution of LexBFS:
(In each step, we highlight vertices, that could be explored next, green. We highlight the vertex, we actually explore (arbitrary choice) next, red. Previously explored vertices are highlighted in blue)

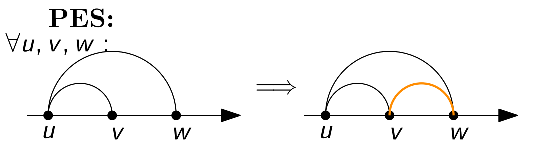


We claim:

Theorem 3.9:
LexBFS returns as PES of $G \iff G$ is chordal.

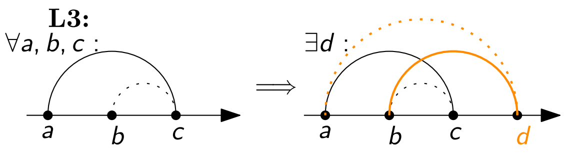
Before we prove this theorem, we rephrase the definition of a PES:

σ is a PES of $G = (V, E) \iff$ for all $u, v, w \in V$ with $u <_\sigma v <_\sigma w$ and $uv, uw \in E$, we have $vw \in E$,



We also note that, if σ is a vertex ordering obtained by LexBFS, then it satisfies the following property:

L3:
For all $a, b, c \in V$ with $a <_\sigma b <_\sigma c$ and $ac \in E$ and $bc \notin E$, there exists a $d \in V$ such that $c <_\sigma d$ and $ad \notin E$ and $bd \in E$.



This is true, because for any such a, b, c, b was selected by LefBFS before a, even though a is adjacent to c and b is not. This can only be true, if b had something in it's label that has a higher value then c and is not contained in the label of a. This 'something', must have been added by a vertex d that satisfied teh requirments,

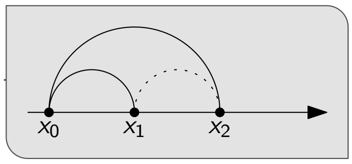
Theorem 3.9:
LexBFS returns as PES of $G \iff G$ is chordal.

Recall:

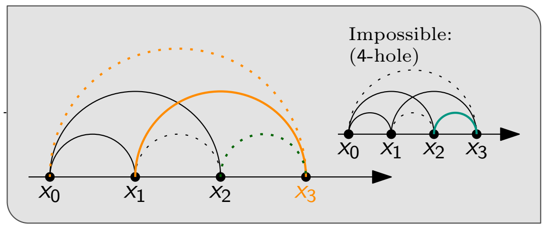
Lemma X (proven):
If G has a PES, then G is chordal.

Proof of Theorem 3.9:

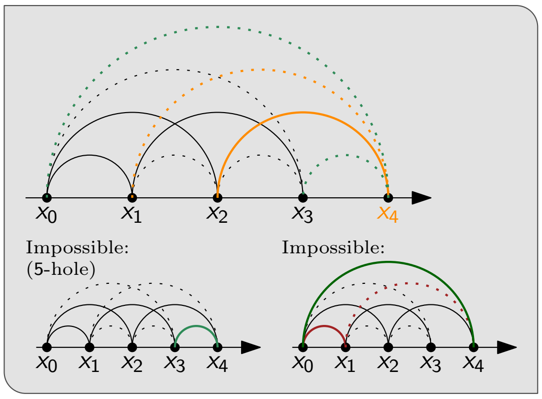
- $\Longrightarrow$: is trivial, by Lemma X
- $\Longleftarrow$: We assume $\sigma :=$ LexBFS(G) is not a PES, even though G is chordal. We will then show the existence of a t-hole oin G, which is a contradction,
- As σ is not a PES, there exist vertices $x_0 <_\sigma x_1 <_\sigma x_2$ that don't satisfy the PES-property, i.e. $x_0x_1, x_0x_2 \in E, x_1x_2 \notin E$.
- W.l.og. let x_2 be rightmost, i.e. there exist no $\bar{x}_2 >_\sigma x_2$ that has the same relation to x_0 and x_1.



- We apply L3 on this triple to obtain a vertex $x_3 >_\sigma x_2$ with $x_1x_3 \in E$, $x_0x_3 \notin E$. (Again, w.l.o.g. let x_3 be rightmost)
- We note that $x_2x_3 \notin E$, as otherwise $(x_0, x_1, x_3, x_2, x_0)$ would be a 4-hole.



- We can now, apply L3 on x_1, x_2, x_3 to get a vertex $x_4 <_\sigma x_3$ with $x_1x_4 \notin E$ and $x_2x_4 \in E$. Again, w.l.og. let x_4 be rightmost.
- We gain $x_3x_4 \notin E$, as otherwise, $(x_0, x_1, x_3, x_4, x_2, x_0)$ would be a 5-hole.
- We gain $x_0x_4 \notin E$, as otherwise x_0, x_1, x_4 would also be a triple that does not satisfy the PES-property. This is a contradiction, as x_2 was chosen to be the rightmost vertex with that property.



- We can repeat this argument indefinetly, to find vertices x_i, $i > 4$ such that x_i is only adjacent to x_{i-2} and not to the vertices $x_0, \dots, x_{i-3}, x_{i-1}$.
- However, as G only has finitely many vertices it is not possible that this process repeats indefinetly (Contradiction).

(done)

We now can detect chordal graphs by:

1. Let $\sigma := LexBFS(G)$.
2. Check whether σ is a PES.

Theorem 3.9 shows the correctness of this algorithm. We will now show that this detection algorithm can be implemented in time $O(|V| + |E|)$.

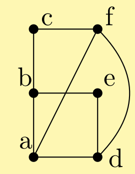
1. Efficient Implementation of LexBFS:

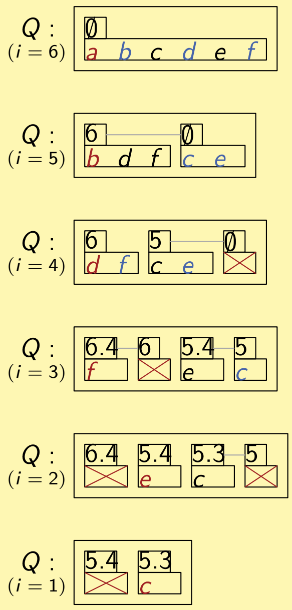
- We implement LexBFS by using a queue Q of lists.
- In each step, every unexplored vertex should be contained in exactly one list in Q.
- A list in Q contains all vertices that have the same label.
- Q orders the lists lexicographically according to the labels inside the lists,
- In each step, we pick a vertex v from the first list L in Q. $(O(1))$
- We can then update Q, by removing v from L and splitting all lists that have a vertex adjacent to v into two separate lists. $(O(\deg v))$

This implementation has runtime

$$O(|V|) + \sum_{v \in V} O(1 + \deg v) = O(|V| + |E|)$$

Example:
(Compare to previous page)

$G =$ 



Pseudocode:

```
for $w \in \mathrm{Adj}(v)$ not numbered do
    if Flag(Set($w$)) = false then
        insert new set $S$ before Set($w$) into $Q$;
        Flag(Set($w$)) ← true; add Set($w$) to FixList;
    end if
    $S$ ← set before Set($w$) in $Q$;
    remove $w$ from Set($w$); add $w$ to $S$;
    Set($w$) ← $S$;
end for
for $S \in$ FixList do
    Flag($S$) ← false;
    if $S$ empty then
        remove $S$ from $Q$;
    end if
    remove $S$ from FixList;
end for
```

Algorithm 2 : Update step in LexBFS

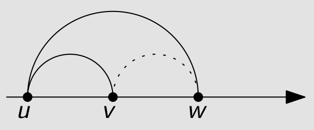
2. Efficient Verification of PES:

Note: The terms "next neighbor", "evil" and "fundamentally evil" on this page were not defined in the lecture.

Problem:
Given a graph G and a vertex ordering σ, decide whether σ is a PES of G.

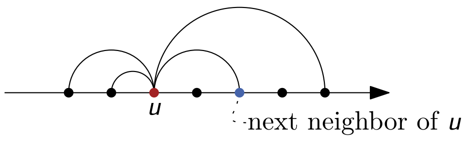
We recall that σ is a PES of $G = (V, E)$, iff. it contains no *evil triples*, where we call a triple of vertices (u, v, w) *evil*, if $u <_\sigma v <_\sigma w$ and $uv, uw \in E$ and $vw \notin E$.

Evil Triple:



This gives us a *naive algorithm* for this problem: Simply check for all triples of vertices, if they are evil. This has runtime $\Theta(n^3)$. We can improve this by observing that we don't actually have to check all possible triples:

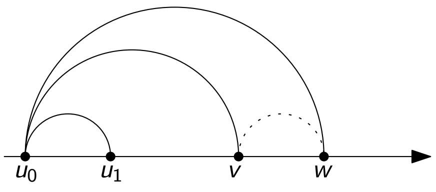
For a vertex u, we call the leftmost neighbor of u that is to the right of u (according to σ), the *next neighbor* of v. We call a triple (u, v, w) *fundamentally evil (f.e.)*, if (u, v, w) is evil and v is the next neighbor of u.



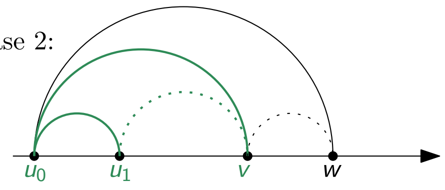
Lemma E:
If σ is a PES of $G \iff (G, \sigma)$ contains no *fundamentally evil* triples.

Proof:

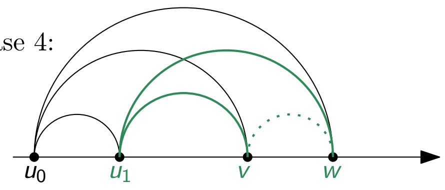
- $\Longrightarrow$: If σ is a PES, it cannot contain any evil triple and hence also no fundamentally evil triples.
- $\Longleftarrow$: By Contraposition: We assume σ is not a PES and show the existence of a fundamentally evil triple.
- As σ is not a PES, there exists an evil triple (u_0, v, w).
- Let u_1 be the next neighbor of u_0. We distinguish cases:
  - Case 1: $u_1 = v \implies (u_0, u_1 = v, w)$ is a f.e. triple. (done)
  - Case 2: $u_1 v \notin E \implies (u_0, u_1, v)$ is a f.e. triple. (done)
  - Case 3: $u_1 w \notin E \implies (u_0, u_1, w)$ is a f.e. triple. (done)
  - Case 4: Otherwise, $u_1 v, u_1 w \in E$. Now, (u_1, v, w) is evil. We note that u_1 is closer to v (according to their ordering in σ), than u_0. We can repeat the argument to find $u_2, u_3, \ldots$ that come closer to v and form a evil triple with (v, w). This cannot repeat indefinetly, hence at some point we must fall into case 1 to 3. (done)

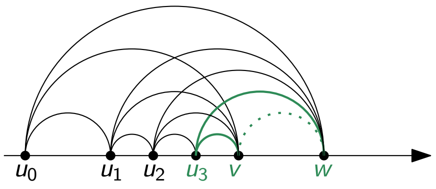


Case 2:



Case 4:





So now, to verify if σ is a PES, it suffices to check the existence of fundamentally evil triples. This speeds the naive algorithm up to a runtime of $\Theta(n^2)$. However, we can do even better (see Algorithm 3 for pseudo code):

- We iterate through the vertices from left to right.
- If we process a vertex u and find it*s next neighbor v, any neighbor w of u might form a f.e. triple (u, v, w). Note that (u, v, w) is f.e. iff $vw \notin E$.
- Importantly, instead of checking if $vw \notin E$ immediatly, we give the vertex v the responsebilty to check this, when it is its turn to be processed.
- To make this communication from u to v efficiently, $Adj(u)$ is stored as a doubly-linked list and v has a doubly-linked list $A(v)$ that stores all it's 'responsibilities'.
- Doubly-connected lists can be concatenated in $O(1)$.
- When it is the turn of v, it can simply check if any f.e. triples (u, v, w) exist, by checking $A(v) - Adj(v) = \emptyset$ (in time $O(\deg v + |A(v)|)$).

```
Input  : graph $G = (V, E)$, vertex ordering $\sigma$.
Output : true, if $\sigma$ PES, false otherwise.

1  for each vertex $v$ do $A(v) \leftarrow \emptyset$;
2  for $i \leftarrow 1$ to $n - 1$ do
3      $v \leftarrow \sigma(i)$;
4      $X \leftarrow \{x \in \mathrm{Adj}(v) \mid \sigma(v) < \sigma(x)\}$;
5      if $X = \emptyset$ then go to line 8;
6      $u \leftarrow \mathrm{argmin}\{\sigma(x) \mid x \in X\}$;
7      add $X - \{u\}$ to $A(u)$;
8      if $A(v) - \mathrm{Adj}(v) \neq \emptyset$ then
9          return false;
10     end if
11 end for
12 return true;
```

Algorithm 3 : Test for perfect elimination scheme

```
Input  : lists $\mathrm{Adj}(v)$, $A(v)$.
Output : true, if $A(v) - \mathrm{Adj}(v) \neq \emptyset$, false otherwise.

for $w \in \mathrm{Adj}(v)$ do $\mathrm{Test}(w) \leftarrow$ true;
for $w \in A(v)$ do
    if $\mathrm{Test}(w) =$ false then
        return true;
    end if
end for
for $w \in \mathrm{Adj}(v)$ do $\mathrm{Test}(w) \leftarrow$ false;
return false;
```

Algorithm 4 : Test for $A(v) - \mathrm{Adj}(v) \neq \emptyset$ in line 8

The correctness of this algorithm follows from Lemma E. The running time consists of precomputations, handeling of a vertex 'in the position of u' and handeling of a vertex 'in the position of v'. In total:

$$O(|V|) + \sum_{u \in V} \deg u + \sum_{v \in V} (\deg v + |A(v)|) = O(|V| + |E|).$$

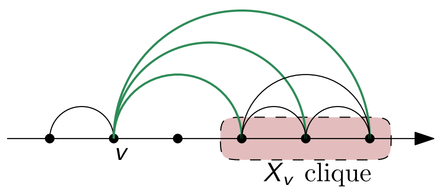
Corollary 3.12:
Chordal graphs can be detected in time $O(|V| + |E|)$.

Algorithms on Chordal Graphs

We will show that we can compute $\omega(G), \alpha(G), \chi(G)$ and $\kappa(G)$ efficiently, if G is a chordal graph.
We first note that we only have to find two algorithms, as $\omega(G) = \chi(G)$ and $\alpha(G) = \kappa(G)$. We show: Algorithm 5 can be used to compute $\omega(G)$ and $\chi(G)$; Algorithm 6 can be used to compute $\alpha(G)$ and $\kappa(G)$.

```
Input  : chordal graph $G = (V, E)$.
Output : clique $C$ and coloring $\phi$.
1  compute with LexBFS a PES $\sigma$ of $G$;
2  $C \leftarrow \emptyset$, $\phi \leftarrow 0$;
3  for $i \leftarrow n$ to $1$ do
4      $v \leftarrow \sigma(i)$;
5      $X_v \leftarrow \mathrm{Adj}(v) \cap \{\sigma(i+1), \dots, \sigma(n)\}$;
6      $\phi(v) \leftarrow \min(\mathbb{N} - \{\phi(w) \mid w \in X_v\})$;
7      if $|C| < |X_v + \{v\}|$ then
8          $C \leftarrow X_v + \{v\}$;
9      end if
10 end for
11 return $C$ and $\phi$;
```

Algorithm 5 : Compute $\omega(G)$ and $\chi(G)$



Theorem:
Algorithm 5 computes a clique C and a coloring ϕ of a chordal graph G such that $|C| = \omega(G)$ and ϕ uses $\chi(G)$ colors.

Proof:

- C is a clique:
  - X_v is a clique as σ is a PES.
  - $X_v + \{v\}$ is a clique as all $u \in X_v$ are adjacent to v by definition.
- ϕ is a coloring
  - Note that $\phi(v)$ is set for each vertex once and is never changed after that.
  - Consider two adjacent vertices $u, v \in V$.
  - Either $u \in X_v$ or $v \in X_u$ holds (w.l.o.g. the former)
  - Now, $\phi(v) \in \mathbb{N} - \{\phi(w) \mid w \in X_v\}$ and hence $\phi(v) \neq \phi(u)$.
- C and ϕ are both optimal.
  - For each $v \in V$: $\phi(v) \leq |X_v| + 1$.
  - For each $v \in V$: $|X_v| + 1 \leq |C|$ — true for all graphs
  - Thus, $\chi(G) \leq \max_{v \in V} \phi(v) \leq |X_v| + 1 \leq |C| \leq \omega(G) \leq \chi(G)$.
  - Hence, $\chi(G) = \max_{v \in V} \phi(v) = |C| = \omega(G)$ (done)

Theorem:
Algorithm 5 can be implemented in time $O(|V| + |E|)$.

Proof: Iteration for each vertex v takes time $O(|Adj(v)|)$. (Line 6 can be done quick by 'crossing out' taken colors in an array that is reused in all iterations.) (done)

```
Input  : chordal graph $G = (V, E)$.
Output : independent set $U$ and clique cover $\psi$.
1  compute with LexBFS a PES $\sigma$ of $G$;
2  $U \leftarrow \emptyset$, $\psi \leftarrow 0$;
3  for $i \leftarrow 1$ to $n$ do
4      $v \leftarrow \sigma(i)$, $X_v \leftarrow \mathrm{Adj}(v) \cap \{\sigma(i+1), \dots, \sigma(n)\}$;
5      if $\psi(v) = 0$ then
6          $U \leftarrow U + \{v\}$;
7          for $w \in X_v + \{v\}$ do
8              $\psi(w) \leftarrow |U|$;
9          end for
10     end if
11 end for
12 return $U$ and $\psi$;
```

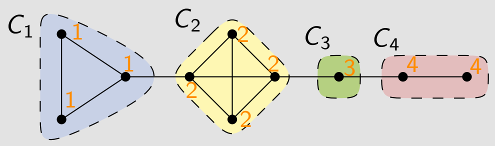
Algorithm 6 : Compute $\alpha(G)$ and $\kappa(G)$

Theorem:
Algorithm 6 computes an i-set U and a clique cover ψ of a chordal graph G such that $|U| = \alpha(G)$ and ψ uses $\kappa(G)$ cliques.

Proof:

- U is an i-set.
  - Assume for the sake of contradiction there exist adjacent u, v in U. W.l.o.g. $u <_\sigma v$.
  - v was added to U in step $i := \sigma^{-1}(v)$. Hence, $\psi(v) = 0$ in step i.
  - In step $j := \sigma^{-1}(u) < i$, in line 8, $\psi(v)$ is set to a value greater 0, as v is adjacent to u by assumption. (Contradiction).
- ψ is a clique cover.
  - For all $i \geq 1$, $\{w \in V \mid \psi(w) = i\} = X_v + \{v\}$ is a clique for some v (line 7 and 8).
  - Every vertex is covered, as all vertices with no assigned number in their iterations are handeled in line 5 to 10.
- U and ψ are both optimal.
  - Note that $\kappa(G) \leq \max_{v \in V} \psi(v) = |U| \leq \alpha(G) \leq \kappa(G)$. (done)

Note:
ψ assign all vertices in the same clique the same number. Similarly to ϕ assigning vertices with the same color the same number.
Example:



clique cover C_1, C_2, C_3, C_4 represented by ψ.

Theorem:
Algorithm 6 can be implemented in time $O(|V| + |E|)$.

Proof:
Similarly to proof for Algorithm 5.

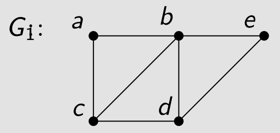
Chordal Graphs as Intersection Graphs

In this final note about chordal graphs, we show that chordal graphs can be described as intersection graphs on subtrees of trees:

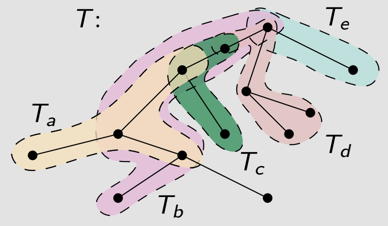
Definition:
We call a graph $G = (V, E)$ a *tree intersection graph*, if there exists a tree T and for each $v \in V$ a subtree T_v of T such that for all $u, v \in V$,

$$uv \in E \iff V(T_u) \cap V(T_v) \neq \emptyset.$$

Example:



G_1 is tree intersection graph, as:



We will show that chordal graphs are exactly the tree intersection graphs. Before that, we will observe some properties of tree intersections:

Definition:
A family of sets $\{S_i\}_{i \in I}$ (indexed by a set I) is said to have the *Helly-Property* iff. for all subsets $J \subseteq I$,

$$\forall i, j \in J : S_i \cap S_j \neq \emptyset \implies \bigcap_{i \in J} S_i \neq \emptyset.$$

Proposition 3.13:
If T is a tree and $\mathcal{X}$ is the set of subtrees of T, then $\mathcal{X}$ satisfies the Helly-Property.

This proposition was given without proof.

Given this proposition, we can now show:

Theorem 3.14:
Let $G = (V, E)$ be any graph. The following statements are equivalent:

(i) G is chordal.

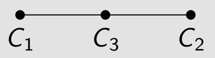
(ii) G is a tree intersection graph.

(iii) There exists a tree T that has vertices $\mathcal{K}$, where $\mathcal{K}$ is the set of maximal cliques of G, such that for any $v \in V$, the subgraph $T[\mathcal{K}_v]$ of T induced by $\mathcal{K}_v$ is connected (and hence a subtree), where $\mathcal{K}_v$ is the set of all maximal cliques in G that contain the vertex v.

G_1 also satisfies statement (iii) from Theorem 3.14, as:

$$\mathcal{K} = \{\underbrace{\{a, b, c\}}_{C_1 :=}, \underbrace{\{b, d, e\}}_{C_2 :=}, \underbrace{\{b, c, d\}}_{C_3 :=}\}$$

T ::



$$\left.\begin{array}{l} \mathcal{K}_a = \{C_1\} \\ \mathcal{K}_b = \{C_1, C_2, C_3\} \\ \mathcal{K}_c = \{C_1, C_3\} \\ \mathcal{K}_d = \{C_2, C_3\} \\ \mathcal{K}_e = \{C_2\} \end{array}\right\}$$ all induce connected graphs in T.

Proof:
(iii) $\implies$ (ii):

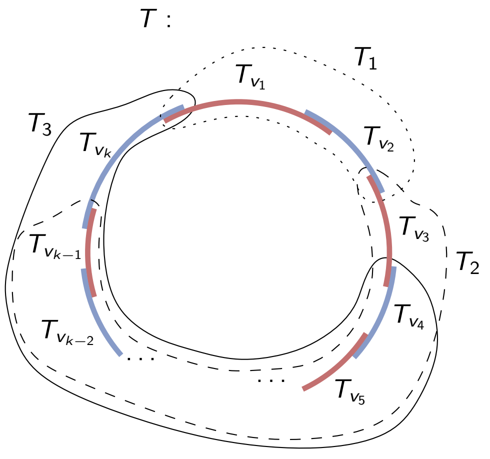
- Assume there exists a tree T with the properties from (iii).
- We set $T_v := T[\mathcal{K}_v]$ for any $v \in V$ and observe for all $u, v \in V$:

$$\begin{aligned} uv \in E &\overset{\text{Let } C \text{ be the maximal clique containing } u \text{ and } v.}{\iff} \exists C \in \mathcal{K} : u, v \in C \iff \exists C \in \mathcal{K} : C \in \mathcal{K}_u \wedge C \in \mathcal{K}_v \\ &\iff \mathcal{K}_u \cap \mathcal{K}_v \neq \emptyset \iff V(T_u) \cap V(T_v) \neq \emptyset. \end{aligned}$$

- Hence, G is a tree intersection graph.

(ii) $\implies$ (i):

- Assume (ii) holds, i.e. there exist subtrees T_v for any $v \in V$ with $uv \in E \iff T_u \cap T_v \neq \emptyset$.
- Let $C = (v_1, \ldots, v_k)$ be any cycle in G with $k \geq 4$.
- We define $T_1 := T_{v_1} \cup T_{v_2}$, $T_2 := T_{v_3} \cup \cdots \cup T_{v_{k-1}}$ and $T_3 := T_{v_4} \cup \cdots \cup T_{v_k}$
- Note that T_1, T_2 and T_3 are connected (and thus subtrees of T) and intersect pairwise. (note e.g. that $T_1 \cap T_2 \supseteq T_{v_2} \cap T_{v_3} \neq \emptyset$, as $v_2 v_3 \in E$.)
- Hence, by the Helly Property, there exists a $x \in \bigcap_{i=1,2,3} V(T_i)$. In particular, $x \in V(T_1) = V(T_{v_1}) \cup V(T_{v_2})$. We distinguish:
  - *Case 1:* $x \in V(T_{v_1})$: As also $x \in V(T_2)$, there exists $i \in \{3, \ldots, k-1\}$ with $x \in V(T_{v_i})$. Thus, $x \in V(T_{v_1}) \cap V(T_{v_i}) \neq \emptyset$. Hence, $v_1 v_i \in E$ is a chord.
  - *Case 2:* $x \in V(T_{v_2})$: Similarly, we find a chord $v_2 v_i$ with $i \in \{4, \ldots, k\}$.
- Hence, G is chordal.



(i) $\implies$ (iii):

- We show this claim by induction over the size n of $V(G)$.
- If $n = 1$, then the claim holds trivially.
- Let $n \geq 2$. As G is chordal, there exists a simphicial vertex $v \in V(G)$.
- Let $G' := G - v$. By induction hypothesis a tree T' exists that satisfies the property from (iii) for G'. Let $A := Adj(v) + \{v\}$.
- We distinguish cases:
- Case 1: $Adj(v)$ is a maximal clique in G'.
  - There is a vertex with name $Adj(v)$ in T'.
  - Let T be the tree obtained by renaming $Adj(v)$ to A in T'.
  - T is tree that satisfies (iii) for G.
- Case 2: $Adj(v)$ is a non-maximal clique in G'.
  - There exists a (not necessarily unique) maximal clique X in G' with $X \supseteq Adj(v)$.
  - Note that A will be a maximal clique in G.
  - Let T be the tree obtained by appending a vertex with name A to the tree T' that is only connected to the vertex with name X.
  - Again, one can verify that T satisfies the conditions from (iii).

(done)

This finishes the chapter about chordal graphs!

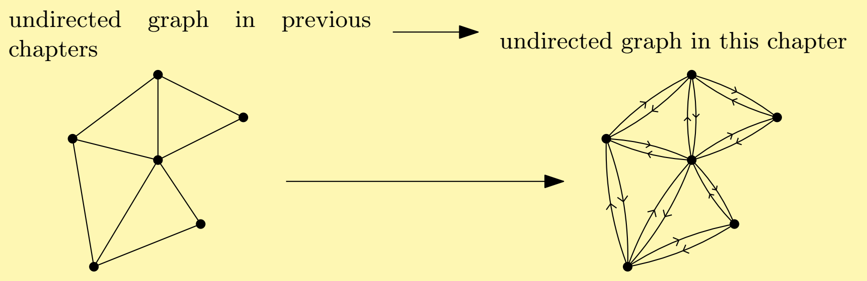
Comparability Graphs

Definition
We call a relation $\sqsubset$ on $A \times A$ a *(strict) order*, if

- $\sqsubset$ is reflexive: $\forall a \in A : a \not\sqsubset a$
- $\sqsubset$ is transitive: $\forall a, b, c \in A : a \sqsubset b \wedge b \sqsubset c \implies a \sqsubset c$

If for $a, b \in A$, neither $a \sqsubset b$ nor $b \sqsubset a$, we say that a and b are incomparable (under $\sqsubset$) and write $a || b$.

Notation:
In this chapter, all graphs considered are directed by default. So, for two vertices $u, v \in V$, we have $uv \neq vu$. Also we have no edges of the form uu. We call a graph $G = (V, E)$ *undirected*, iff. $\forall u, v \in V : uv \in E \implies vu \in E$.


undirected graph in previous chapters
undirected graph in this chapter

Intuitively, comparability graphs are graphs that illustrate a strict ordering (see Example to the right). This motivates the following definitions:

Definition
An *orientation* of an (undirected) graph $G = (V, E)$ is a subset of edges $F \subseteq E$ such that for all $uv \in E$ it holds that $uv \in F \iff vu \notin F$.
Furthermore, for any subset $F \subseteq E$, we define

$$F^{-1} := \{vu \mid uv \in F\}$$

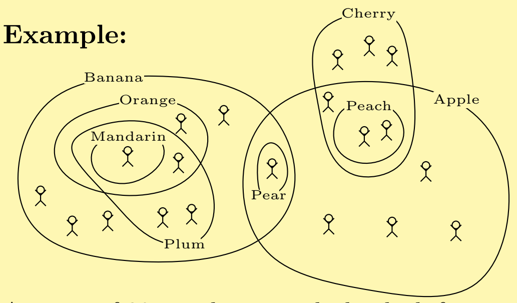
and the *symmetric closure*

$$\hat{F} := F \cup F^{-1}.$$

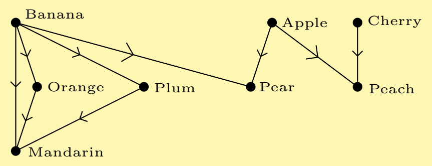
Definition:
We call an orientation F transitive, iff. $\forall a, b, c \in V : ab \in F \wedge bc \in F \implies ac \in F$.

Definition:
We call an undirected graph G a *comparability graph*, if there exists a transitive orientation F of G.

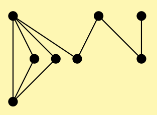
Example:

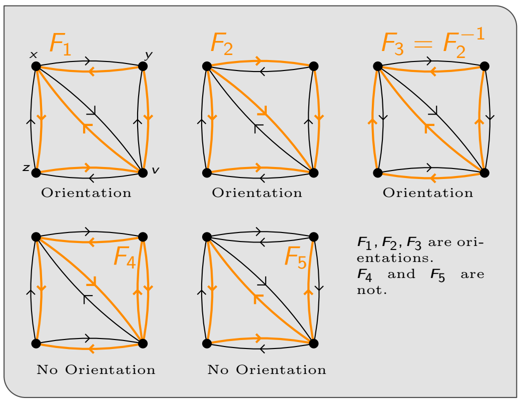


A group of 20 people were asked, which fruits they liked. The answers of the people are illustrated above (e.g. there are 6 people that like cherries). We say that fruit a is better than another fruit b, if all people that like b also like a (we write $b \sqsubset a$). Observe:
Peach $\sqsubset$ Cherry, Mandarin $\sqsubset$ Banana, but Cherry || Apple.
This ordering can also be illustrated as a directed graph:



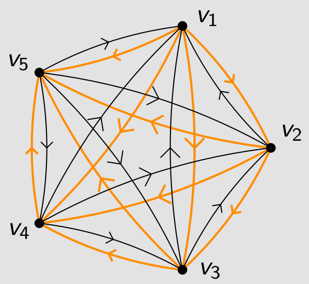
If we remove the directions of the edges, we obtain the graph to the right. We want to call graphs that can be obtained that way *comparability graphs*



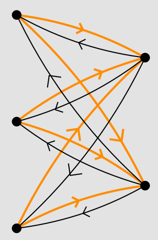


In the example above, F_2 and F_3 are transitive. F_1 is not transitive, as $yx \in F_1$ and $xz \in F_1$, but not $yz \in F_1$.

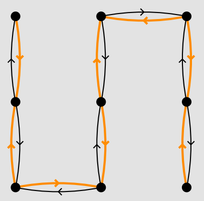
Lets look at some examples of comparability graphs:



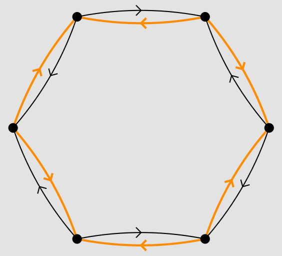
K_n for all n:
$v_i v_j \in F \iff i < j$.



Bipartite Graphs (in particular $K_{n,m}$ for all n, m):
$V = A \dot\cup B$,
$ab \in F \iff a \in A, b \in B$.



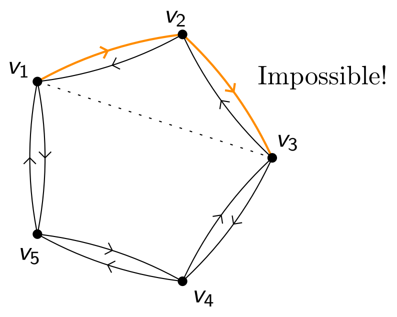
P_n for all n:
orient edges alternating.



C_n for **even** n:
Orient edges alternatingly.

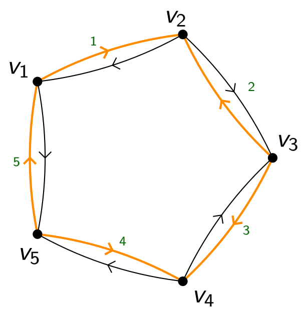
However, not all graphs are comparability graphs:

C_t is no comparability graph, for **odd** $t \geq 5$.



Proof:

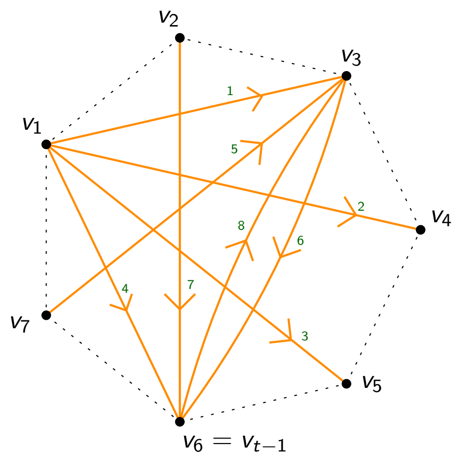
- Assume F was a transitive orientation of $C_t = (v_1, \dots, v_t)$.
- W.l.o.g. Let $v_1 v_2 \in F$. (if this is not the case, we could instead consider F^{-1})
- Now, if $v_2 v_3 \in F$, then by transitivity also $v_1 v_3 \in F$ (Contradiction, as $v_1 v_3 \notin E$). Hence, $v_3 v_2 \in F$.
- Now, if $v_4 v_3 \in F$, then by transitivity also $v_4 v_2 \in F$ (Contradiction, as $v_4 v_2 \notin E$). Hence, $v_2 v_4 \in F$.
- By repeating this argument, we note that the edges must be oriented alternatingly in clockwise and anticlockwise direction.
- As t is odd, this is impossible. (done)



$\overline{C_t}$ is no comparability graph, for **odd** $t \geq 5$.

Proof:

- Note that $\overline{C_5} = C_5$. Thus, we only consider $t \geq 7$.
- Again, assume there were a transitive orientation F of $C_t = (v_1, \dots, v_t)$.
- W.l.o.g. $v_1 v_3 \in F$.
- We observe (similar to the proof above) by repeatedly applying the transitivity of F, the following facts:
  - $v_1 v_3 \in F \wedge v_3 v_4 \notin E \implies v_1 v_4 \in F$.
  - $v_1 v_4 \in F \wedge v_4 v_5 \notin E \implies v_1 v_5 \in F$.
  - ...
  - (For the remaining observations, see the green numbers in the sketch to the right, which show in which order, we decided that an edge must be in F.)
- In the end we get a contradiction, as we have shown that $v_3 v_{t-1} \in F$ and $v_{t-1} v_3 \in F$. (done)



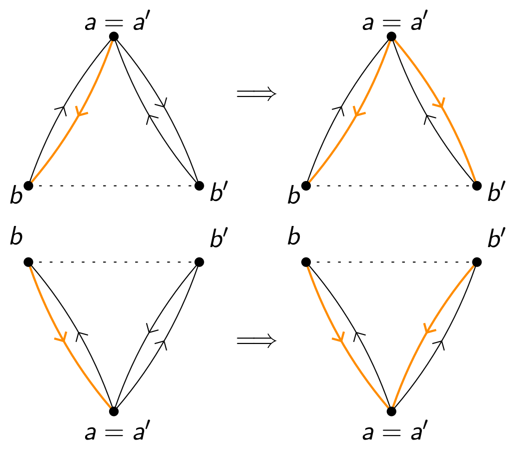
Corollary: G is a comparability graph $\implies$ G is perfect

Proof:

- Let G be non-perfect.
- By the SPGT, G has either C_t or $\overline{C_t}$ as an induced subgraph with $t \geq 5$ odd.
- By the two observations above on this page, it follows that G is not a comparability graph. (done)

We will later show this corollary without use of the SPGT, which we have not proven yet.

In the proofs concerning C_t and $\overline{C_t}$, we used a methode of 'cascading implications'. We will make this observation implicit:



Observation:
Let F be a transitive orientation of undirected $G = (V, E)$. If $ab \in F$ and $a'b' \in E$, where $a = a'$ and $bb' \notin E$ (or $b = b'$ and $aa' \notin E$), then $a'b' \in F$.

Proof: Assume not, so instead of $a'b' \in F, b'a' \in F$. Then with transitivity and $a'b = ab \in F$ it follows $b'b \in F$. This is not possible, a $b'b \notin E$. (done)

Based on this observation, we define:

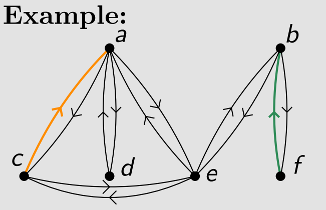
Definition:
For an undirected graph $G = (V, E)$, we define a binary relation Γ on the edge set of a graph by: For $ab \in E$, $a'b' \in E$, we have $ab \,\Gamma\, a'b'$ iff.

- Either $a = a'$ and $bb' \in E$,
- or $b = b'$ and $aa' \in E$.

Definition:
We define the binary relation Γ^* as the transitive hull of Γ. i.e. we set $ab\Gamma^* a'b'$ iff. there exist $a_1b_1, \dots, a_kb_k \in E$ such that $a_1b_1 = ab$, $a_kb_k = a'b'$ and for all i, $a_ib_i \,\Gamma\, a_{i+1}b_{i+1}$.

Note that Γ^* is reflexive, symmetric and transitive and hence, a equivalence relation on the edge set E of a graph G. Thus, Γ^* splits E into equivalence classes. The set of all equivalence classes under Γ^* is denoted by $\mathcal{I}(G)$.

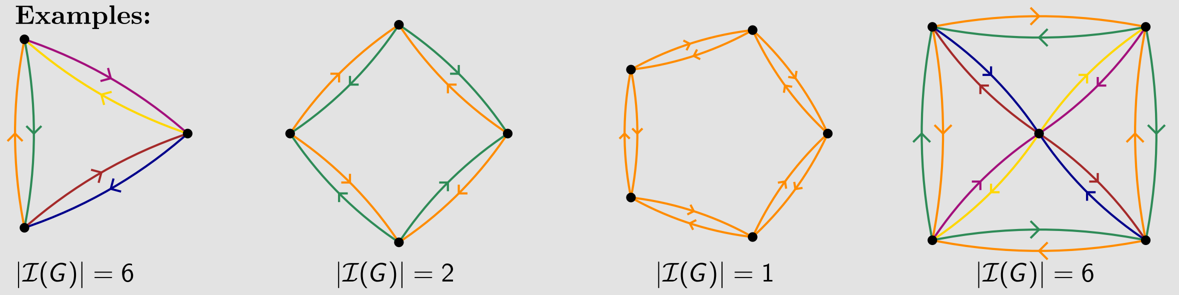
Example:



Note that $ca \,\Gamma\, da$, $da \,\Gamma\, ea$, $ea \,\Gamma\, eb$, $eb \,\Gamma\, fb$. Hence, $ca \,\Gamma^*\, fb$.

Hint: you can see the Γ / Γ^* relation as walking with your fingers. Start by setting one finger on c and one finger on a. Observe how your fingers walk, when you walk down the Gamma-chain ca, da, ea, eb, fb. A finger may always jump to another vertex that is non-adjacent to the current vertex.

Examples:



We note that if G is a comparability graph, than $|\mathcal{I}(G)$ is even. This is because in that case equivalence classes 'come in pairs'. We will prove this fact later. To do so, we first need to make some notes:
If $A \in \mathcal{I}(G)$, then also $A^{-1} \in \mathcal{I}(G)$. Recall that $\hat{A} := A \cup A^{-1}$. We call $\hat{A}$ a color class of G. Let $\hat{\mathcal{I}}(G)$ be the set of all color classes of G. We prove:

Theorem 4.1:
If $A \in \mathcal{I}(G)$ and F is a transitive orientation of G, then either $F \cap \hat{A} = A$ or $F \cap \hat{A} = A^{-1}$.

In other words: Either all of A is contained in F or all of A^{-1} is contained in F.

Proof:

- Let $ab \in \hat{A}$, w.l.o.g. $ab \in A$.
- Case 1: $ab \in F$
  - Let $cd \in A$ be arbitrary.
  - Then $ab \,\Gamma^*\, cd$.
  - Hence, $cd \in F$ is forced.
  - Thus, $A \subseteq F$.
  - As $F^{-1} \cap F = \emptyset$, we have $F \cap A^{-1} = \emptyset$.
  - Hence, $F \cap \hat{A} = (F \cap A) \cup (F \cap A^{-1}) = A \cup \emptyset = A$. (done)
- Case 2: $ab \notin F$.
  - Then, $ba \in F$ (and $ba \in A^{-1}$)
  - Remainder is analog to Case 1. (done)

Corollary:
If G is a comparability graph, then $A \cap A^{-1} = \emptyset$ for all implication classes $A \in \mathcal{I}(G)$.

This Corollary gives us a necessary condition for comparability graphs. Surprisingly, this condition is even sufficient. This will be part of a main result of this chapter, which we will prove in Theorem 4.7.

Proof:
Assume $A \cap A^{-1} \neq \emptyset$. Then there exists $ab \in A \cap A^{-1}$, which also implies $ba \in A \cap A^{-1}$. As G is a comparability graph, there exists a transitive orientation F of G. Let $B := F \cap \hat{A} \in \{A, A^{-1}\}$ and note that $ab, ba \in B = F \cap \hat{A} \subseteq F$.
(Contradiction)

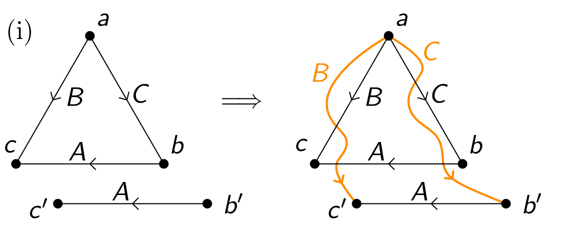
The following lemma will be useful for upcoming proofs:

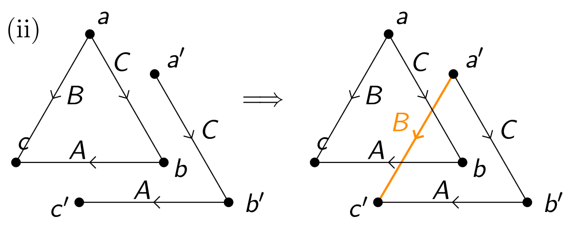
Triangle Lemma:
Let $G = (V, E)$ be an undirected graph and $A, B, C \in \mathcal{I}(G)$ such that $A \neq B$ and $A \neq C^{-1}$. Furthermore, let $ab \in C$, $ac \in B$ and $bc \in A$. Then,

(i) $b'c' \in A \implies ab' \in C, ac' \in B$.

(ii) $b'c' \in A \wedge a'b' \in C \implies a'c' \in B$.

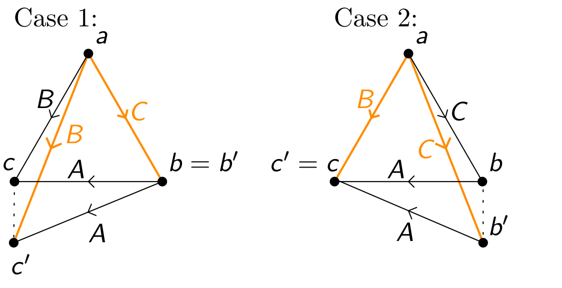
Note that only $A \neq B$ and $A \neq C^{-1}$ are required. However, it is possible e.g. to set $A = C$, $A = B^{-1}$, $b' = b$, $b' = a$, ...





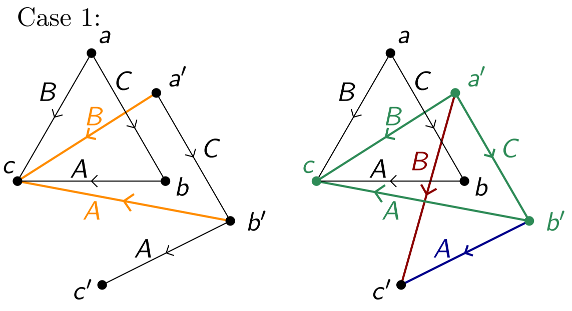
Proof of Part (i):

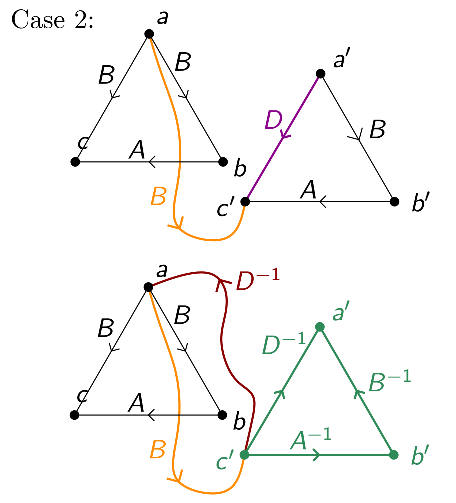
- As $b'c' \in A$ and $bc \in A$, we have $bc\,\Gamma^*\,b'c'$ by definition.
- W.l.o.g. we consider the case $bc\,\Gamma\,b'c'$ (if the Gamma chain has more then one step, we can apply this step multiple times).
- There are two cases: Either $b = b'$ and $cc' \notin E$ or $bb' \notin E$ and $c = c'$.
- Case 1: $b = b'$ and $cc' \notin E$.
  - First note that $ac' \in E$, as otherwise $C^{-1} \ni ba\,\Gamma\,bc' = b'c' \in A$ (but we assumed $A \neq C^{-1}$).
  - Now, $ac'\,\Gamma\,ac \in B$ and hence $ac' \in B$.
  - Also note that $ab' = ab \in C$. (done)
- Case 2: $c = c'$ and $bb' \notin E$.
  - Analogously, but using $A \neq B$. (done)



Proof of Part (ii):

- We distinguish two cases:
- Case 1: $B \neq C$
  - Note that $ab\,\Gamma^*\,a'b'$ and again we assume again w.l.o.g. $ab\,\Gamma\,a'b'$.
  - With a similar case distinction, as in Part (i), we show that $a'c \in B$ and $b'c \in A$. (here we use $A \neq C^{-1}$ and $B \neq C$)
  - Now, note that a', b' and c satisfy the requierments of the Triangle Lemma and we can apply Part (i) on the edge $b'c'$ to obtain $a'c' \in B$, (done)
- Case 2: $B = C$
  - If $a'c' \notin E$, then $A \ni b'c'\,\Gamma\,b'a' \in B^{-1} = C^{-1}$ and hence $A = C^{-1}$ (impossible). Thus, $a'c' \in E$.
  - Let $D \in \mathcal{I}(G)$ be chosen such that $a'c' \in D$. For the sake of contardiction, we assume $D \neq B$.
  - By directly applying Part (i) on $b'c'$, we obtain $ac' \in B$.
  - However, we can also apply Part (i) on the triangle a', b', c' with implication classes A^{-1}, D^{-1} and B^{-1} (This is possible as $B^{-1} \neq A^{-1}$ and $B^{-1} = C^{-1} \neq A = (A^{-1})^{-1}$): By $ba \in B^{-1}$ follows $c'a \in D^{-1}$.
  - This however implies $ac' \in D$, which together with $ac' \in B$ contadicts the assumption of $D \neq B$. (done)



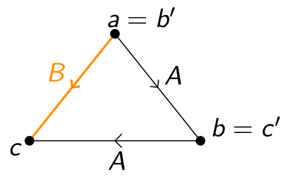


As shown in Theorem 4.1, there is a connection between implication classes and a transitive orientation F of a graph. We have shown that F is always a union of implication classes. This motivates us to study the transitivity of implication classes:

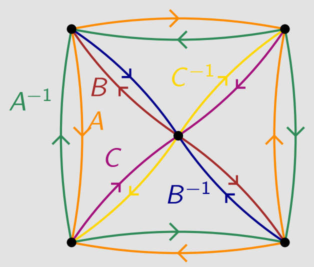
Theorem 4.4
Let $G = (V, E)$ be a (undirected) graph and $A \in \mathcal{I}(G)$. Then either $A = A^{-1}$ or $A \cap A^{-1} = \emptyset$ **and** A, A^{-1} are transitive.

Proof:

- We first show that either $A = A^{-1}$ or $A \cap A^{-1} = \emptyset$.
  - Assume $A \cap A^{-1} \neq \emptyset$. Hence, there exists $ab \in A \cap A^{-1}$.
  - Let $cd \in A$ be arbitrary and note that $cd\,\Gamma^*\,ab\,\Gamma^*\,ba\,\Gamma^*\,dc$ (note that we used in the middle step that ab and ba are both contained in A (and A^{-1} too)).
  - Now, $dc \in A$ and thus, $cd \in A^{-1}$.
  - This proves $A \subseteq A^{-1}$ and as $|A| = |A^{-1}|$, this already implies $A = A^{-1}$.
- It remains to show that $A \cap A^{-1} = \emptyset$ implies that A is transitive. (Given this, teh transitivity of A^{-1} follows immediatly.)
  - Let $a, b, c \in V$ such that $ab \in A$, $bc \in A$, but $ac \notin A$.
  - Note that $ac \in E$, as otherwise $A \ni ab\,\Gamma^*\,cb \in A^{-1}$ (but we know $A \neq A^{-1}$).
  - Let $B \in \mathcal{I}(G)$ such that $ac \in B$. We will show that $A = B$.
  - F.s.o.c. assume $A \neq B$.
  - Note that we can apply the Triangle Lemma on this situation (with $C = A$). We set $b'c' := ab$ and apply Part (i) to obtain that $B \ni ac' = ab$.
  - But we know that $ab \in A$ and hence, $A = B$ (Contradiction).



This theorem might motivate us to consider the following approach for finding a transitive orientation F of a graph G: For all $A, A^{-1} \in \mathcal{I}(G)$, we simply decide arbitrarily whether we include A or A^{-1} into F.
However: This approach fails, as the example to the right shows. It seems like some implication classes are dependent on each other.



$F_1 := A + B + C$ is a **non**-transitive orientation. But $F_2 = A^{-1} + B + C$ is transitive.

Nevertheless, a small variation of this approach works as intended and calculates a transitive orientation T of G, if isuch orientation exists (see Algorithm 7):

- Set $T := \emptyset$ and $i = 1$.
- Choose any implication class B_i of G and set $T := T + B_i$ (if $B_i \neq B_i^{-1}$).
- Remove $\hat{A}$ from G, set $i := i + 1$ and repeat previous step.

Input : undirected graph $G = (V, E)$.
Output : transitive orientation T, if it exists.

```
1  $T \leftarrow \emptyset$;
2  $i \leftarrow 1$; $E_i \leftarrow E$;
3  while $E_i \neq \emptyset$ do
4      choose $x_i y_i \in E_i$ arbitrarily;
5      determine implication class $B_i$ of $E_i$ containing $x_i y_i$;
6      if $B_i \cap B_i^{-1} \neq \emptyset$, then
7          return "G is no comparability graph";
8      end if
9      add $B_i$ to $T$;
10     $E_{i+1} \leftarrow E_i - \hat{B}_i$;
11     $i \leftarrow i + 1$;
12 end while
13 return $T$;
```

Algorithm 7 : Recognition of comparabilty graphs

We will now prove the corretness of Algorithm 7, which is not trivial, as removing edges from G may change the structure of the implication classes. We start by analysing the classes $B_1, \dots, B_k$ by Algorithm 7:

Definition:
For a (undirected) graph $G = (V, E)$, we call $[B_1, \dots, B_k]$ a G-*decomposition*, if $\hat{B}_1 + \dots + \hat{B}_k = E$ and for all i, $B_i \in \mathcal{I}(\hat{B}_i + \dots + \hat{B}_k)$.

Note that the $B_1, \dots, B_k$ computed by Algorithm 7 form a G-decomposition. The correctness of the algorithm follows from the following theorem:

Remark:
Algorithm 7 can be implemented in time $\mathcal{O}(\Delta(G)|E| + |V|)$

Theorem 4.7
Let G be an (undirected) graph. The following statements are equivelent:

(i) G is a comparability garph.

(ii) $A \cap A^{-1} = \emptyset$ for all $A \in \mathcal{I}(G)$.

(iii) For all G-decompositions $[B_1, \dots, B_k]$ it holds that $B_i \cap B_i^{-1} = \emptyset$ for all $i \in [k]$.

Before we can show this theorem, we need to establish a connection between $\mathcal{I}(G)$ and $\mathcal{I}(G - \hat{A})$ for some implication class $A \in \mathcal{I}(G)$:

Theorem 4.6
Let $G = (V, E)$ be an (undirected) graph and $A \in \mathcal{I}(G)$ and $D \in \mathcal{I}(G - \hat{A})$. Exactly one of the two following statements is true:

(i) $D \in \mathcal{I}(G)$ and $A \in \mathcal{I}(G - \hat{D})$.

(ii) $D = B + C$ for some $B, C \in \mathcal{I}(G)$ and $\hat{A}, \hat{B}, \hat{C}$ contain a rainbow triangle.

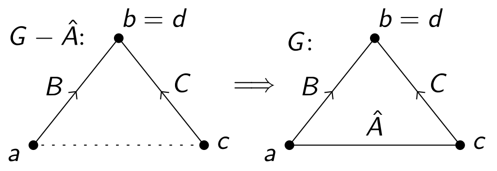
Definition:
We call $\{a, b, c\}$ a *rainbow triangle* in pairwise distinct color classes $\hat{A}, \hat{B}, \hat{C} \in \hat{\mathcal{I}}(G)$, if $bc \in \hat{A}, ac \in \hat{B}$ and $ab \in \hat{C}$.

Remark:
Removing edges from G (so adding non-edges) does not remove any Γ relations between any two remaining edges (see definition) i.e. it can only happen that some edgesin $G - \hat{A}$ that were not in a Γ-relation in G, now have a Γ -relation in $G - \hat{A}$. This means that all $D \in \mathcal{I}(G)$ are either already contained in $\mathcal{I}(G)$ or are a union of multiple implication classes in $\mathcal{I}(G)$.
The surprising part of Theorem 4.6 is that D can only be a union of *at most* 2 implication classes.

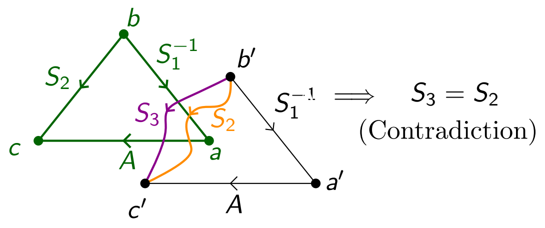
Proof:

- As explained in the remark to the right, D can be expressed as $D = S_1 + \dots S_k$ with $S_i \in \mathcal{I}(G), i \in [k]$.
- Case 1: $k = 2$.
  - Call $B := S_1, C := S_2$ and note $D = B + C$.
  - Then there exist edges $ab \in B$, $cd \in C$ such that $ab \,\Gamma\, cd$ in $G - \hat{A}$, but $ab \not\Gamma cd$ (in G).
  - By the definition of Γ this implies $ac \notin E - \hat{A}$ and $b = d$ or $bd \notin E - \hat{A}$ and $a = c$. (w.l.og the former).
  - Now, $\{a, b, c\}$ is a rainbow triangle in $\hat{A}, \hat{B}, \hat{C}$. (done)
- Case 2: $k \geq 3$
  - As in Case 1, we obtain a rainbow triangle $\{a, b, c\}$ in $\hat{A}, \hat{S}_1, \hat{S}_2$ and a rainbow triangle $\{a', b', c'\}$ in $\hat{A}, \hat{S}_1, \hat{S}_3$.
  - With the Triangle Lemma (ii) and some handwaving follows that $\hat{S}_3 = \hat{S}_2$. (The handwaving is 'necessary', as we would have to consider all the 2^6 possible 'orientations' of $ab, bc, a'b', \dots$, i.e. the choices of having $ab \in S_1$ or $ab \in S_1^{-1}$, ... The figure to the right shows the application of the Triangle Lemma for oen such 'orientation') (Contradiction)
- Case 3: $k = 1$ i.e. $D = S_1 \in \mathcal{I}(G)$.
  - We want to show that $A \in \mathcal{I}(G - \hat{D})$. F.s.o.c. we assume $A \notin \mathcal{I}(G - \hat{D})$.
  - By Case 2 it can only happend that A merged together with *exactly one* other implication class $X \in \mathcal{I}(G)$ i.e. $X + A \in \mathcal{I}(G - \hat{D})$.
  - Now, by Case 1, there exists s rainbow triangle $\{a, b, c\}$ in $\hat{D}, \hat{A}, \hat{X}$.
  - But in this case, D would merge with X or X^{-1} in $G - \hat{A}$. (Consider what happens, if you remove the edge $ac \in \hat{A}$ from the rainbow triangle, see figure to the right).
  - This is a contradiction, as $D \in \mathcal{I}(G - \hat{A})$.
  - Hence, $A \in \mathcal{I}(G - \hat{D})$. (done)

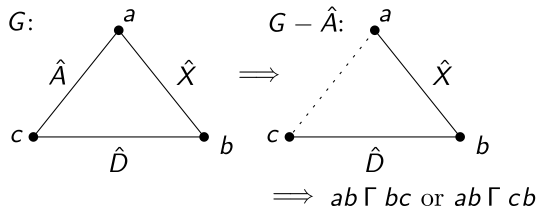
Case 1:



Case 2:



Case 3:



We can now prove Theorem 4.7 and hence the correctness of Algorithm 7.

Theorem 4.7
Let G be an (undirected) graph. The following statements are equivelent:

(i) G is a comparability graph.

(ii) $A \cap A^{-1} = \emptyset$ for all $A \in \mathcal{I}(G)$.

(iii) For all G-decompositions $[B_1, \dots, B_k]$ it holds that $B_i \cap B_i^{-1} = \emptyset$ for all $i \in [k]$.

Proof:

- (i) $\Longrightarrow$ (ii): Done, as shown in Theorem 4.1

- (ii) $\implies$ (iii)
  - Let G be a graph that satisfies (ii) and $[B_1, \dots, B_k]$ be any G-decomposition.
  - We will show by induction over k that $B_i \cap B_i^{-1}$ for all $i \in [k]$.
  - $k = 1$: Now, $B_1 \in \mathcal{I}(G)$ and hence by (ii), $B_1 \cap B_1^{-1} = \emptyset$.
  - $k \geq 2$:
    - Again, $B_1 \cap B_1^{-1} = \emptyset$ by (ii).
    - $[B_2, \dots, B_k]$ is a $(G - \hat{B}_1)$ -decomposition.
    - If we can verify property (ii) for $G - \hat{B}_1$, we can apply the induction hypothesis and are done.
    - Hence, let $D \in \mathcal{I}(G - \hat{B}_1)$ be arbitrary (we will show $D \cap D^{-1} = \emptyset$).
    - We apply Theorem 4.6 and fall in one of the two cases:
    - Case 1: $D \in \mathcal{I}(G)$ and we are done, as (ii) holds for G.
    - Case 2: $D = B + C$ for some $B, C \in \mathcal{I}(G - \hat{B}_1)$.
      - Then (recall that we assume (ii) for G):

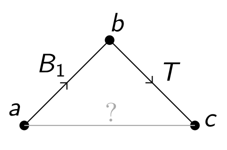
$$\begin{aligned} D \cap D^{-1} &= (B + C) \cap (D + C)^{-1} = (B + C) \cap (B^{-1} + C^{-1}) \\ &= \underbrace{(B \cap B^{-1})}_{=\emptyset \text{ by (ii)}} + \underbrace{(B \cap C^{-1})}_{=\emptyset \text{ by } (*)} + \underbrace{(C \cap B^{-1})}_{=\emptyset \text{ by } (*)} + \underbrace{(C \cap C^{-1})}_{=\emptyset \text{ by (ii)}} \\ &= \emptyset. \end{aligned}$$

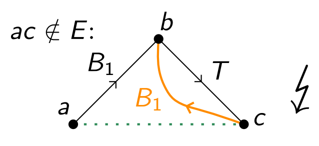
$(*)$: Assume e.g. $B \cap C^{-1} \neq \emptyset$, then $B = C^{-1}$ and hence $\hat{B} = \hat{C}$. But, by Theorem 4.6 $\hat{B}_1, \hat{B}$ and $\hat{C}$ must contain a rainbow triangle and hence be distinct (by definition), (Contradiction).

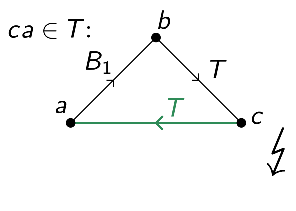
    - In both cases $D \cap D^{-1} = \emptyset$ (for all $D \in \mathcal{I}(G - \hat{B}_1)$). Hence, $G - \hat{B}_1$ satisfies property (ii). By induction hypothesis this implies (iii) for $G - \hat{B}_1$, which finishes this part.

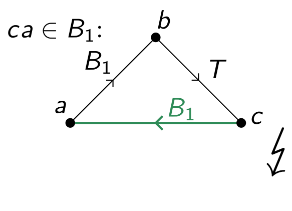
- (iii) $\implies$ (i)
  - We assume (iii) to be true for G.
  - Let $[B_1, \dots, B_k]$ be **any fixed** G-decomposition. (such decomposition exists always, as we can run Algorithm 7 but deleting line 6,7,8.)
  - We will show by induction over k that (i) also holds for G i.e. that G is a comparability graph.
    - $k = 1$: As $B_1 \in \mathcal{I}(G)$ satisfies $B_1 \cap B_1^{-1}$ by (iii), we get that B_1 is transitive by Theorem 4.4. Hence, B_1 is a transitive orienattion of G.
    - $k \geq 2$:
      - By induction hypothesis, we know that $G - \hat{B}_1$ is a comparability graph and hence has a transitive orientation T,
      - Note that $B_1 + T$ is a orientationn of G. We will now show that $B_1 + T$ is **transitive**:
      - Let $ab, bc \in B_1 + T$ be arbitrary. We show that this implies $ac \in B_1 + T$.
      - This is obviously true, if ab, bc are both in B_1 or both in T, as B_1 and T are both transitive.
      - Hence, w.l.o.g. we are only interested in the case where $ab \in B_1$ and $bc \in T$.
      - Note that $ac \in E$, as otherwise $ab \,\Gamma\, cb$ and hence $cb \in B_1$. (Contradicition)
      - Now, if $ca \in T$, then T would be non-transitive (as $bc, ca \in T$ but $ba \in B_1^{-1} \neq T$).
      - But, if $ca \in B_1$, then B_1 would be non-transitive (as $ca, ab \in B_1$ but $cb \in T^{-1} \neq B_1$).
      - Hence, $ca \notin T + B_1$ and (as $T + B_1$ is an orientation), $ac \in T + B_1$.

(done)









Algorithms on Comparability Graphs

We will now find algorithms for computing $\omega(G)$ $(= \chi(G))$ and $\alpha(G)$ $(= \kappa(G))$.

Computing $\omega(G)$ and $\chi(G)$:
See Algorithm 8. We show the correctness, but first recall:

Definition:
A *topological ordering* of an oriented graph (V, F) is a mapping $\sigma : V \to [n]$ such that $uv \in F \implies \sigma(u) < \sigma(v)$.

Theorem:
Algorithm 8 computes an optimal coloring and a maximal clique of G, if G is a comparability graph.

Proof:

- First note that line 5 is well defined, as if $wv \in F$, we $\sigma(w) < \sigma(v)$ and hence already defined $h(w)$, when we process the iteration of v.
- h is obviously a proper coloring, as for all $uv \in E$ (w.l.og. $uv \in F$), we have
$$h(v) = 1 + \max\{h(w) \mid wv \in F\} \geq 1 + h(u) > h(u).$$
- We also note that the value of $h(w)$ decreases always by one in each iteration of the lines (9,10,11,12). Hence, $C = \{w_\chi, w_{\chi-1}, \dots, w_2, w_1\}$ is well-defined.
- As $w_i w_{i+1} \in F$ for all i and by transitivity, we have that $w_i w_j \in F$ for all $i < j$. Hence, C is a clique of G.
- So, we have shown:
$$\chi(G) \leq \chi := \max_{v \in V} h(v) = |C| \overset{\text{(always)}}{\leq} \omega(G) \leq \chi(G)$$
(done)

Computing $\alpha(G)$ and $\kappa(G)$:
This is a little more involved and no pseudocode is provided. Before we consider comparability graphs in general, we focus on a subclass of graphs, namely bipartite graphs. We observe a connection to matchings and vertex covers:

Definition:
We call $M \subseteq E$ a *matching* of $G = (V, E)$, if for all $v \in V$ there exists *at most one* $e \in M$ such that e and v are incident, We call v covered, if such an e exists.

Observation 1:
For any graph $G = (V, E)$, if M is a matching of G, then $\kappa(G) \leq |V| - |M|$.

Proof Sketch: Cover all covered vertices by the 2-cliques (edges) in M and cover the remaining vertices by 1-cliques.

Definition:
We call a set $S \subseteq V$ a *vertex cover* of $G = (V, E)$, if for all $e \in E$ there exists a $v \in S$ such that v and e are incident.

Observation 2:
For any graph $G = (V, E)$, S is a vertex cover of G, iff. $V - S$ is independent. In particular,
$\alpha(G) = |V| - \min\{|S| \mid S \text{ is vtx. cvr.}\}$.

From the two observation follows:
$$|V| - \{|S| \mid S \text{ is vtx. cov. }\} = \alpha(G) \leq \kappa(G) \leq |V| - \{|M| \mid M \text{ is matching}\}.$$
The following theorem shows that all 4 terms are equal if G is bipartite:

Theorem of König:
If G is bipartite, then
$$|V| - \{|S| \mid S \text{ is vtx. cov. }\} = \alpha(G) = \kappa(G) = |V| - \{|M| \mid M \text{ is matching}\}.$$

Proof Sketch: First equality by Observation 2; Second equality by perfectness of bipartite graphs; Third equality can be shown similarly to Observation 1, while noting that $\omega(G) \leq 2$.

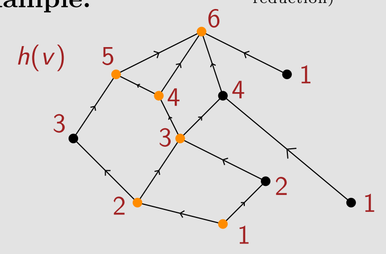
Input : comparability graph $G = (V, E)$.
Output : vertex coloring h and clique C.

```
1  compute transitive orientation $F$ of $G$;
2  compute tological ordering $\sigma$ of $(V, F)$;
3  for $i \leftarrow 1$ to $n$ do
4      $v \leftarrow \sigma(i)$;
5      $h(v) \leftarrow 1 + \max\{h(w) \mid wv \in F\}$;   Note: $\max\{\} := 0$
6      $\chi \leftarrow \max\{\chi, h(v)\}$;
7      $w \leftarrow \mathrm{argmax}\{h(w), h(v)\}$;
8  end for
9  for $i \leftarrow \chi$ to $1$ do
10     $C \leftarrow C + \{w\}$;
11     $w \leftarrow \mathrm{argmax}\{h(v) \mid vw \in F\}$;
12 end for
13 return $h$ and $C$;
```

Algorithm 8 : Compute $\chi(G)$ and $\omega(G)$

Example:

Only non-transitive edges drawn (transitive reduction)

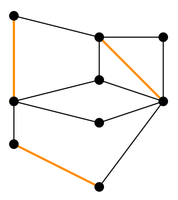


h is a coloring and the orange vertices form a clique.

Remark:
Ignoring line 1, Algorithm 8 runs in time $\mathcal{O}(|V| + |E|)$

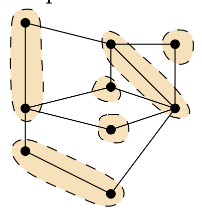
Definition:
An (undirected) graph $G = (V, E)$ is called *bipartite*, if there exists a partition $A + B = V$ of the vertices such that for any edge $xyinE$, we have $|A \cap \{x, y\}| = 1$ and $|B \cap \{x, y\}| = 1$

Matching M:

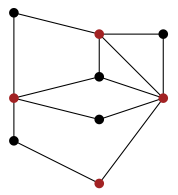


$\Longrightarrow$

Clique Cover:

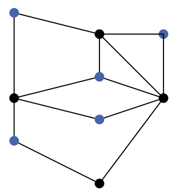


Vertex Cover S:



$\Longleftrightarrow$

Independent Set I:

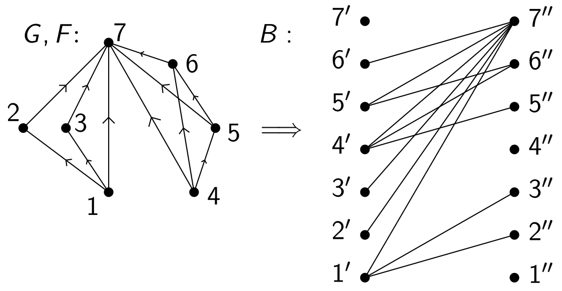


Note:
A minimal clique cover and a maximal matching of G can be computed efficiently (in polynomial time), if G is bipartite. (See e.g. Kuhn's algorithm)

Hence, we know how we can compute $\alpha(G)$ and $\kappa(G)$ efficiently on bipartite graphs.

Surprisingly, this helps us computing $\alpha(G)$ and $\kappa(G)$ on arbitrary comparability graphs.

Let $G = (V, \cdot)$ be a comparability graph and F a transitive orientation of G. We define the auxillary undirected graph B by $V(B) = V' + V''$, where $V' = \{v' \mid v \in V\}$ and $V'' = \{v'' \mid v \in V\}$, and $E(B) = \{v'w'' \mid v, w \in V, vw \in F\}$. Note that B is bipartite.



We note:

Lemma:
For any $k \in \mathbb{N}$, given a clique cover $V_1 + \cdots + V_k$ of G, we can efficiently compute a matching M of B with $|M| = |V| - k$ of B (and vice versa).

Note: If the vertex names are chosen according to a topological ordering of (V, F), all edges in B have positive slope (if drawn as in this example).

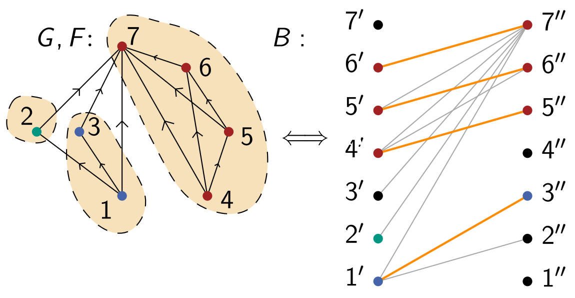
Proof Sketch:
$\Longrightarrow$

- Given $V_1 + \cdots + V_k$.
- For each $i \in [k]$ order V_i according to F, i.e. $V_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,|V_i|}\}$ with $v_{i,a}v_{i,b} \in F$ for all $a < b$.
- Set $M := \{v_{i,a}v_{i,a+1} \mid i \in [k], a \in [|V_i| - 1]\}$.
- Note that M is a matching and
$$|M| = \sum_{i=1}^{k}(|V_i| - 1) = \left(\sum_{i=1}^{k}|V_i|\right) - k = |V| - k.$$

$\Longleftarrow$

- Given matching M of B.
- For any unmatched $v'' \in V$:
  - Create new set $C := \{v\}$.
  - If v' is matched i.e. $v'w'' \in M$ for some (unique) $w \in V$, then add w to C, set $v := w$ and repeat this step.
- All the set created this way together form a clique cover of G. (done)

Clique Cover: Matching:



Lemma:
For any $k \in \mathbb{N}$, given an independent set I with $|I| = |V| - k$ of G, we can efficiently compute a vertex cover S of B with $|S| = k$ and $|\{v', v\} \cap S| \leq 1$ for all $v \in V$ (and vice versa).

This Lemma was phrased differently in the lecture (or is wrong).

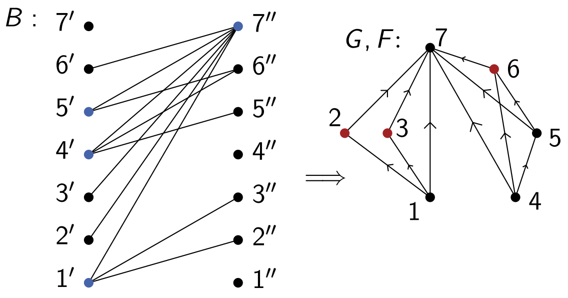
Proof Sketch:
$\Longleftarrow$

- Given vertex cover S of B, we set $I = \{v \in V \mid v' \notin S, v'' \notin S\}$.
- I is an independent set of G, as for any $vw \in E$ (w.l.o.g. $vw \in F$), we know that (as $vw \in E(B)$) either $v' \in S$ or $w'' \in S$. Hence, $v \notin I$ or $w'' \notin I$.

$\Longrightarrow$
unclear (for me)

(done)

Vertex Cover: Independent Set:



TODO: Mention that any minimal vertex cover S of B has $|\{v', v''\} \cap S| \leq 1$.

Those two Lemmas show that we can compute a maximal independent set and a minimal clique cover of a comparability graph by first computing teh auxillary graph B and computing a maximal matching and a minimal vertex cover (which is easy, as B is bipartite). The runtime is dominated by the computation of the transitive orientation F and the maximal matching M.

I am very unsure on whether the last half of this page is correct.

Split Graphs

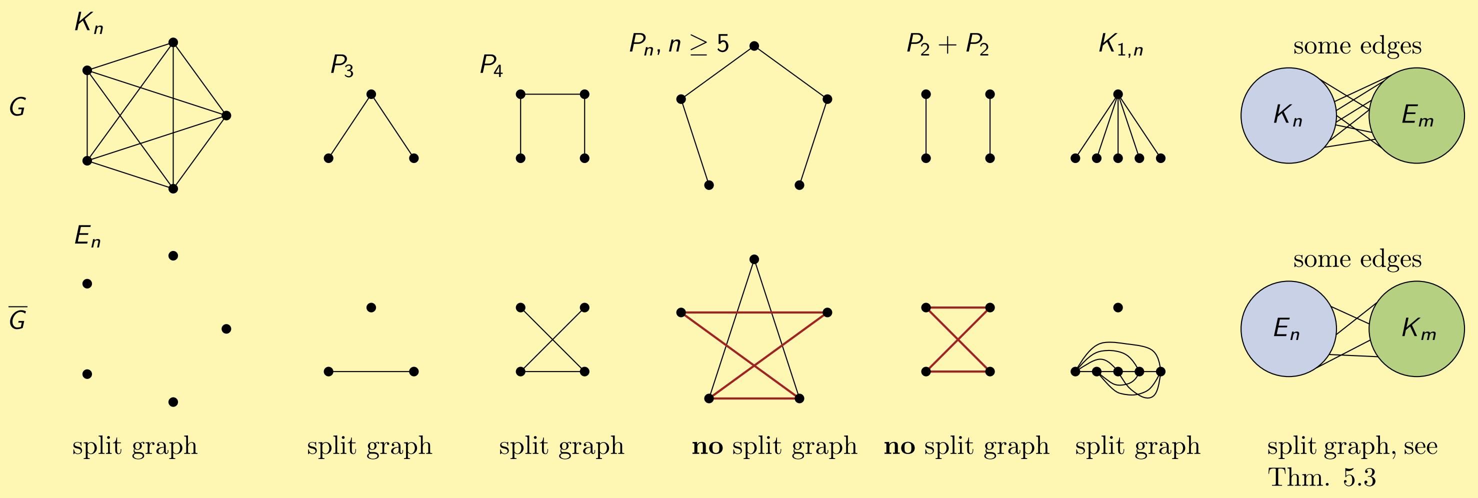
P	$\overline{P}$	C	$\overline{C}$	graph class	
✓				comparability graphs	Chap.4
		✓		chordal graphs	Chap.3
	✓	✓		interval graphs	Chap.7
		✓	✓	split graphs	Chap.5
✓	✓			permutation graphs	Chap.6
✓		✓		cycle-free partial orders	???

Consider the following properties of an (undirected) graph G:
Property P: G is a comparability graph.
Property $\overline{P}$: $\overline{G}$ is a comparability graph.
Property C: G is a chordal graph.
Property $\overline{C}$: $\overline{G}$ is a chordal graph.

We have already observed all these properties in isolation. The remainder of the lecture is about combinations of those properties (see table to the right).

Definition:
A graph G is called a *split graph*, if it satisfies properties C and $\overline{C}$, i.e. if G and $\overline{G}$ are both chordal.

Examples:

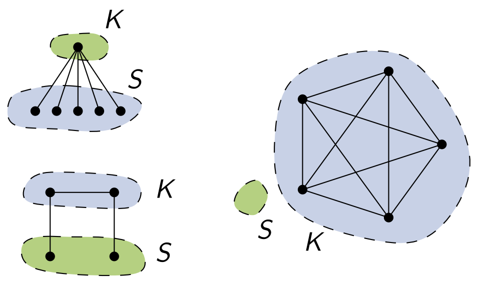


Theorem 5.3:
For every graph $G = (V, E)$ the following are equivalent:

(i) G is a split graph

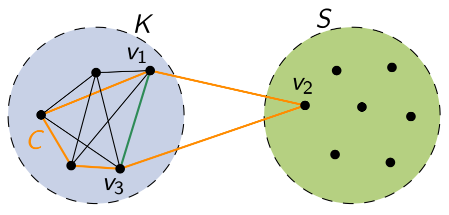
(ii) There exist a clique K and an independent set S such that $V = K + S$.

(iii) G contains no induced copies of C_4, C_5 or $\overline{C}_4$.



Proof (Sketch):
(ii) $\implies$ (i)

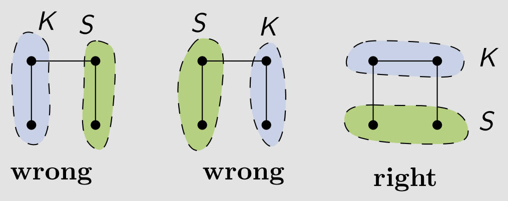
- Assume $V = K + S$, where K is a clique and S is an independent set.
- Let $C = (v_1, \ldots, v_t)$ be any cycle in G with $t \geq 4$. We show that C has a chord.
- If C is completely contained in K, then this is trivially true.
- Hence, we assume there exists a vertex (w.l.o.g. v_2) in C that is contained in S.
- As S is independent, we know that $v_1, v_3 \notin S$.
- Hence, $v_1, v_3 \in K$, which implies that v_1v_3 is a chord.



(i) $\implies$ (iii): trivial
(iii) $\implies$ (ii):

- Let K be a (cardinally) maximum clique of G such that G_S has the least number of edges, where $S := V - K$.
- We assume there to be an edge $xy \in E(G_S)$ and lead this to a contradiction.
- The proof (which was shown in the lecture) is long and boring, containing many (13) case distinctions.

Note:
Simply choosing any maximum clique K and setting $S := V - K$ does not always work, as can be seen for P_4:



Permutation Graphs

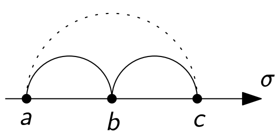
Definition:
We call a graph G a *permutation graph*, if it satisfies properties P and $\overline{P}$, i.e. if G and $\overline{G}$ are both comparability graphs.

Observation:
A graph G is a comparability graph iff. there exists a vertex ordering σ of G such that (σ, G) does not contain M_1 (for one direction, we can choose σ as a topological ordering of a tranitive orientation of G).
Similarly, $\overline{G}$ is a comparability graph iff. there exists a vertex ordering σ' such that (σ', G) does not contain M_2.
As a consequence, G is a permutation graph iff. there exists a vertex ordering σ that does not contain M_1 and there exists a vertex ordering σ' that does not contain M_2. The interesting part of Theorem 6.A (ii) is that we can select σ and σ' in a way that $\sigma = \sigma'$.

Given a vertex ordering σ of a graph $G = (V, E)$, observe the two following patterns:

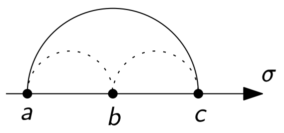
M_1:

$(a, b, c) \in V^3$
$a <_\sigma b <_\sigma c$
$ab, bc \in E, ac \notin E$.



M_2:

$(a, b, c) \in V^3$
$a <_\sigma b <_\sigma c$
$ac \in E, ab, bc \notin E$.



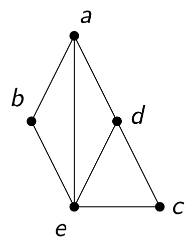
Theorem 6.A:
For every (undirected) graph $G = (V, E)$ the following are equivalent:

(i) G and $\overline{G}$ are comparability graphs (i.e. G is a permutation graph).

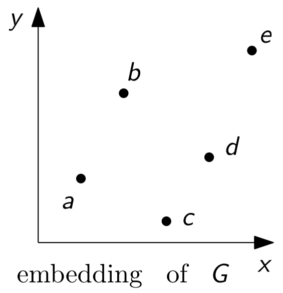
(ii) There exists a vertex ordering σ of G such that σ does not contain M_1 and does not contain M_2.

(iii) There exists an embedding $V \to \mathbb{R}^2$ such that

$$uv \in E \quad \text{if and only if} \quad u_x < v_x \iff u_y < v_y$$



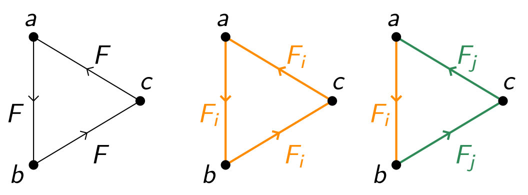
G



embedding of G according to (iii)

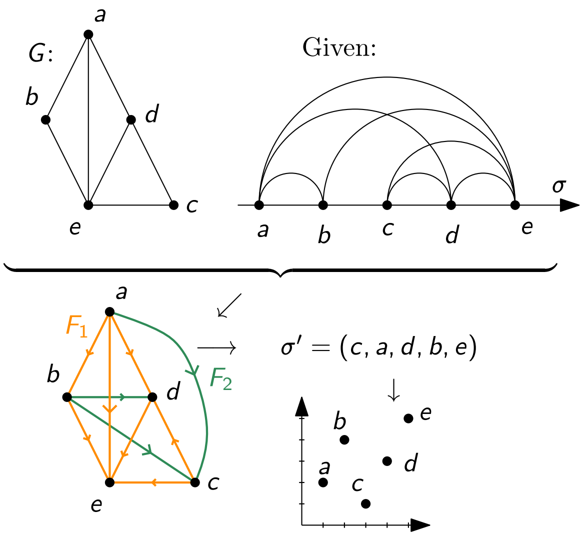
Proof:
(i) $\implies$ (ii)

- Let F_1 (respectively F_2) be a transitive orientation of G (respectively $\overline{G}$).
- Let $F := F_1 + F_2$. Note that F is a orientation of the complete graph on V.
- We show that F is transitive.
  - Assume there exist $ab, bc \in F$ with $ac \notin F$.
  - As $ac \in E(K_n)$ and F is a orientation, this implies $ca \in F$.
  - If there exists an $i \in [2]$ such that $ab, bc, ca \in F_i$, then F_i is obviously non-transitive (Contradiction).
  - Hence, there are $i, j \in [2], i \neq j$ such that one of the edges $\{ab, bc, ca\}$ is in F_i (w.l.o.g. ab) and the other two (w.l.og. bc and ca) are contained in F_j.
  - Then $ba \notin F_j$, but $bc, ca \in F_j$ (Contradiction, as F_j is transitive).
- Let σ be a topological ordering of (V, F).
- If σ would contain the pattern M_1 (respectively M_2), this would contradict the transitivity of F_1 (respectively F_2).



(ii) $\implies$ (iii)

- Let σ be a vertex ordering not containing M_1 or M_2.
- Let $F_1 := \{uv \mid uv \in E(G), u <_\sigma v\}$ and $F_2 := \{uv \mid uv \in E(\overline{G}), u <_\sigma v\}$. Note that F_1 and F_2 are transitive orientations of G and $\overline{G}$ respectively.
- We observe again that $F_1 + F_2^{-1}$ is also transitive and hence there exists a topological ordering σ' of $(V, F_1 + F_2^{-1})$.
- We define the embedding $V \to \mathbb{R}^2, v \mapsto (\sigma(v), \sigma'(v))$ and verify that it satisfies the requiered properties.

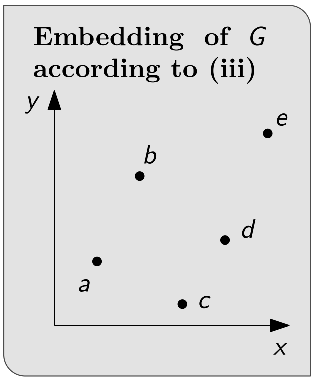


(iii) $\implies$ (i)

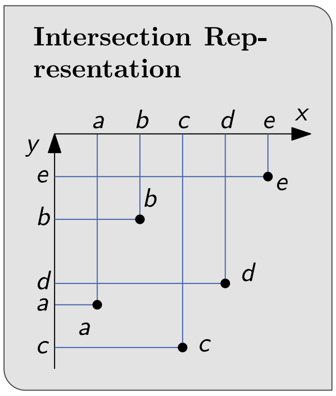
- Let F_1 be the orientation of G, where each edge is oriented from u to v iff. u is to the bottom left of v.
- Note that this "bottom-left" relation is transitive.
- Similarly, we find a transitive orientation F_2 of $\overline{G}$, by orienting an edge from u to v iff. u is to the top left of G.

(done)

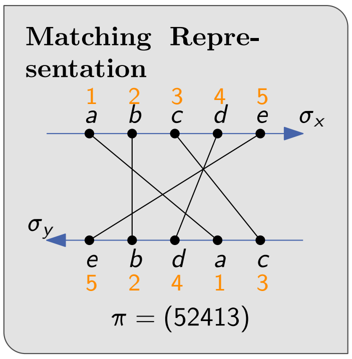
Note: I did not attend the lecture for this page and only looked at the write-ups from previous years.



The 2-dimensional embedding of a permutation graph (as in Theorem 6.A(iii)) also yields another way of representing permutation graphs: We shift the y-axis downwards and rays that go leftwards and upwards to all vertices. This representation of permutaion graphs is called *intersection representation*. In this representation two vertices are adjecent iff. any of their rays intersect.



Alternatively, a permutation graph can be representated in the *matching representation*, where we represent the vertices of G as line segments between two horizontal lines. Tow vertices are adjacent iff. the corresponding line segments intersect. The examples to the right show how you can obtain a matching representation given an intersection representation.



We note that only the order in which the endpoints of the line segments are placed on the horizontal lines is relevant. Hence, a matching representation can be uniquely described by a permutation $\pi : V \to V$ of the vertices. For a given permutation π, we call the graph with matching representation permutation π the *inversion graph* of π and denote it by $G[\pi]$.

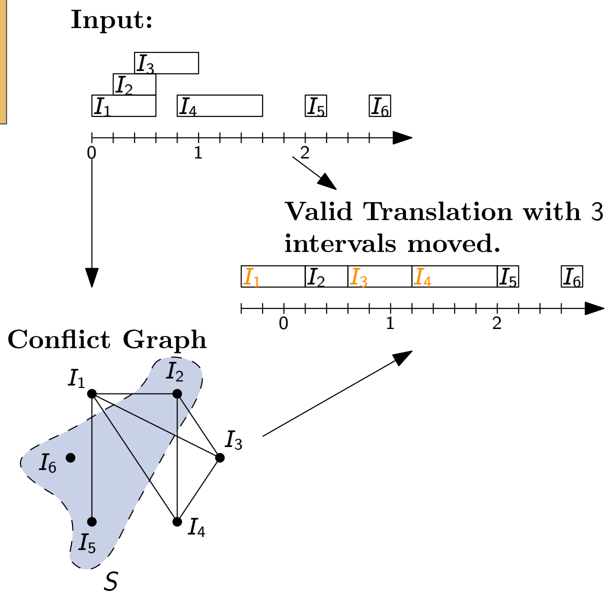
Given π, we can calculate $\chi(G[\pi])$ and $\omega(G[\pi])$ in time $O(|V| + |E|)$. $\alpha(\cdot)$ and $\kappa(\cdot)$ can also be computed efficiently, as the complement of $G[\pi]$ is $G[\pi']$, where π' is π but in reversed order.

As an application of these algorithms, we consider the following problem:

Given: Open intervals $I_1, \dots, I_n$ with $I_i = (x_i, y_i)$ sorted such that $x_1 \leq x_2 \leq \dots \leq x_n$.
Find: Minimal number of translation of the intervals needed such that the intervals remain sorted by x_i and do not intersect pairwise, i.e., $x'_1 \leq x'_2 \leq \dots \leq x'_n$ and $y'_i \leq x'_{i+1}$ for all $i \in [n-1]$.

We say that two intervals I_i and I_j are in conflict iff. it is inevitable to move either I_i or I_j. This is the case, iff. $x_j - y_i \leq \sum_{i<k<j} y_k - x_k$. Let G be the conflict graph with $V(G) = \{I_1, \dots, I_n\}$ and $I_iI_j \in E(G)$ iff. I_i is in conflict with I_j.

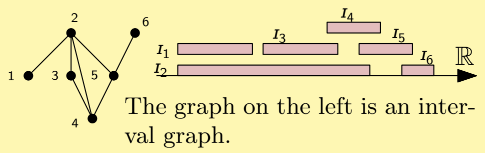
We can show that G is a permutation graph by choosing $\sigma = (I_1, \dots, I_n)$ as in Theorem 6.1(ii). We now want to find a way to translate as few intervalls as possible to resolve those conflicts. This is equivalent to finding a set S of as much intervals as possible that are pairwise not in conflict. The set S is the maximum independent set of G, which can be computed fast as G is a permutation graph. Now, we only have to move the intervals in $V(G) - S$ to resolve the conflicts.



Interval Graphs

Recall:

Definition:
We call a graph $G = (V, E)$ an *interval graph* if for each $v \in V$ there exists an interval I_v such that for all $u, v \in V$: $uv \in E \iff I_u \cap I_v \neq \emptyset$.



The graph on the left is an interval graph.

We have shown in a previous chapter that interval graphs are chordal. We now show some equivalent formulations of interval graphs:

Theorem 7.1
For an (undirected) graph $G = (V, E)$, the following statements are equivalent:

(i) G is an interval graph.

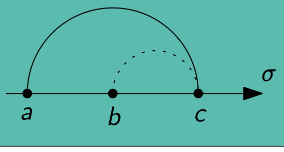
(ii) There exists a vertex ordering σ such that G with σ does not contain the pattern M_3.

(iii) G is chordal and $\overline{G}$ is a comparability graph.

(iv) G does not contain C_4 as an induced subgraph and $\overline{G}$ is a comparability graph.

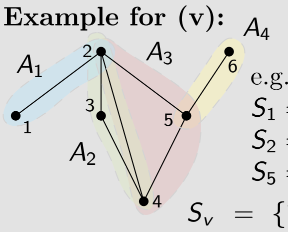
(v) There exists an ordering $A_1 \prec A_2 \prec \cdots \prec A_x$ of the inclusion-maximal cliques of G such that for each vertex $v \in V$, $S_i := \{i \mid v \in A_i\}$ is an interval of $\{1, \ldots, x\}$.

Pattern M_3:
Given vertex ordering σ of $G = (V, E)$, we condider the following pattern:
$(a, b, c) \in V^3$:
$a <_\sigma b <_\sigma c$
$ac \in E$, $bc \notin E$.



Note:
We say nothing about ab.

Example for (v):

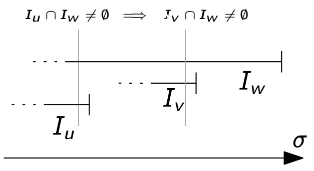


e.g.
$S_1 = \{1\}$, ✓
$S_2 = \{1, 2, 3\}$, ✓
$S_5 = \{3, 4\}$. ✓

$S_v = \{1, 3\}$ would not be allowed.

Proof:
(i) $\implies$ (ii)

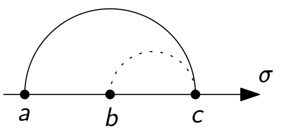
- Let I_u be the corresponding interval of each $u \in V$.
- We sort the vertices from left to right by the right endpoints of their intervals and call the resulting ordering σ. (w.l.o.g. are the endpoints pairwise disjoint)
- Consider $u, v, w \in V$ with $u <_\sigma v <_\sigma w$ and $uv \in E$ (i.e. $I_u \cap I_v \neq \emptyset$).
- We show $vw \in E$ by looking to the sketch to the right.



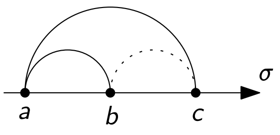
$I_u \cap I_w \neq \emptyset \implies I_v \cap I_w \neq \emptyset$
I_w
I_v
I_u
σ

(ii) $\implies$ (iii)

(ii) $\implies$ $\exists \sigma$: no



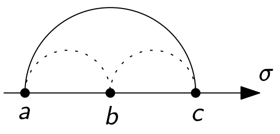
$\implies$ no



$\implies$ G is chordal.

and

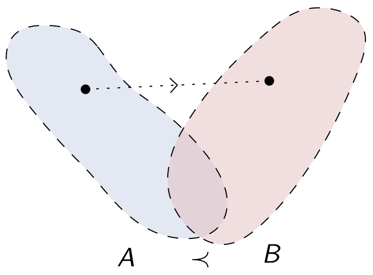
$\implies$ no



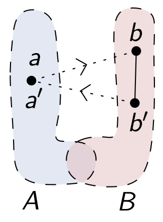
$\implies$ $\overline{G}$ is a comparability graph.

(iii) $\implies$ (iv): trivial
(iv) $\implies$ (v)

- Assume G does not contain C_4 as an induced subraph and $\overline{G}$ has a transitive orientation F.
- We define the linear order $\prec$ on all maximal cliques of G:
  - Let A, B be maximal cliques of G.
  - We set $A \prec B$ iff. there exists a non-edge $e \in F$ oriented from $A - B$ to $B - A$.
- As there always exists a non-edge $e \in E(\overline{G})$ between $A - B$ and $B - A$ (as B and A are maximal), we have either $A \prec B$ or $B \prec A$ (or both).
- $\prec$ is antisymmetric
  - Assume $A \prec B$ and $B \prec A$. Then there exist $a, a' \in A$ and $b, b' \in A$ such that $ab \in F$ and $b'a' \in F$.
  - All three cases ($a = a'$, $b \neq b'$), ($a \neq a'$, $b = b'$) and ($a \neq a'$, $b \neq b'$) lead to a contradiction each (The first and third case are illustrated to the right).

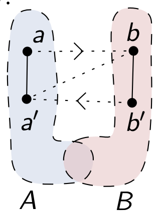


Case 1:
$a = a'$, $b \neq b'$



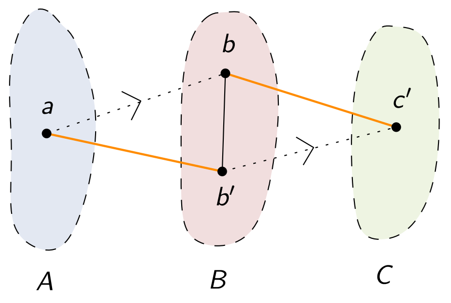
$b'a, ab \in F \implies$
$b'b \in F \subseteq E(\overline{G})$ ↯

Case 3:
$a \neq a'$, $b \neq b'$
Either $ab' \notin E$ or $a'b \notin E$ (wlog the latter), as otherwise $C_4 \subseteq_{\text{ind}} G$.



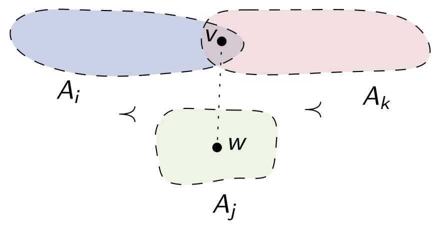
Either $a'b \in F$ or $ba' \in F$. Both contradict transitivity of F. ↯

- $\prec$ is transitive.
  - Let A, B, C be maximal cliques with $A \prec B$ and $B \prec C$.
  - Hence, there exist $ab \in F$ and $b'c' \in F$ with $a \in A, b \in B, b' \in B$ and $c' \in C$. We will show that $ac \in F$.
  - If $b = b'$ or $ab' \notin E$ or $bc' \notin E$, we follows $ac \in F$ immediately by transitivity of F.
  - Thus, we assume $b \neq b'$ and $bc', ac \in E$.
  - As G may not contain C_4 as an induced subgraph, we know that $ac' \notin E$.
  - By the transitivity of F, $c'a \in F$ is impossible and hence, $ac' \in F$.
  - This shows $A \prec B$.



Note that $a \in A$ and $c' \in C$ implies $a \in A - C$ and $c' \in C - A$, as there can be no non-edge from $A \cap C$ to either A or C, as A and C are cliques.

- Hence, we have shown that $\prec$ really is a total order of the maximal cliques of G: $A_1 \prec A_2 \prec \cdots \prec A_x$.
- Now, let $v \in V$ be arbitrary. We want to show that $S_v := \{i \mid v \in A_i\}$ is an interval of $\{1, \dots, x\}$.
- It suffices to show that for all $i < j < k$ with $v \in A_i$ and $v \in A_k$ it follows that $v \in A_j$.
- Assume $v \notin A_j$, Then, as A_j is maximal, we find a $w \in A_j$ such that $vw \notin E$.
- If $vw \in F$, we have $A_k \prec A_j$. $\lightning$
- If $wv \in F$, we have $A_j \prec A_i$. $\lightning$



(v) $\implies$ (i)

- Assume (v) holds.
- For each $v \in V$ we define $I_v := [\min S_v, \max S_v]$.
- Now, for all $v, w \in V$,

$$uv \in E \iff \exists i : v, w \in A_i \iff S_v \cap S_w \neq \emptyset \iff I_v \cap I_w \neq \emptyset.$$

- Hence, G is an interval graph. (done)

AGT Notes © 2025 by Karl Bernhard is licensed under CC BY 4.0. To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/