<u>Einführung - Orga</u> 23.04.25

VL : Übung : frei = 2 : 1 : 1 (Verhältnis)

Prüfung: mündlich ; 3 mögliche Prüfungswochen ; ~ 20 min

Website ; keine Folien → Handschrift ; alte Folien ggf in alten Ilias
⇒ kein Ilias-Kurs , updates auf Website

neuer Raum: Bauingenieur-Bau , Raum 102

Einführung - Fachlich

theory-lecture → graphs and formal proofs, then algorithms on graphs

Def: A Graph $G = (V, E)$ with: V = (finite) vertex set ; $|V| \geq 1$
E = set of edges, $E \subseteq \{ u \times v \mid$
pairs of vertices $\leftarrow \binom{V}{2} = \leftarrow \{ \quad u, v \in V, u \neq v \}$

⇒ undirected simple graphs, no double edges

notation : $\{u,v\} = uv$, hence $uv = vu$

n → usually natural Number $n \in \mathbb{N}$

Important graphs

$[n] = \{1, \dots, n\}$

complete graphs: $n \geq 1$, $K_n = (\{1, \dots, n\}, \binom{[n]}{2})$

$V(K_n) = [n]$ → has n vertices → n number of vertices

$E(K_n) = \binom{[n]}{2}$

examples:

K_1 K_2 K_3 K_4

→ infinite graph family

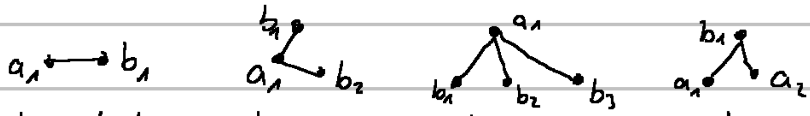
complete bipartite graphs: $n, m \geq 1$

$K_{n,m} = \{\{a_1, \dots, a_n\} \cup \{b_1, \dots, b_m\}, E(K_{nm})\}$

$V(K_{n,m})$

$E(K_{n,m})$

examples:



$K_{1,1}$ ($\sim K_2$) $K_{1,2}$ $K_{1,3}$ $K_{2,1}$ ($\sim K_{1,2}$)

→ infinite graph family

cycles: $n \geq 3$, $C_n = ([n], \{\{i, i+1\} \mid i \in [n-1]\} \cup 1n)$

$V(C_n)$ → n number of vertices

$E(C_n)$

examples:

C_3 C_4 C_5

→ $C_n \approx P_n + 1n$ → circle is path with edge $1n$

paths $n \geq 1$ $P_n = ([n], \{\{i, i+1\} \mid i \in [n-1]\})$

$V(P_n)$

$E(P_n)$

examples:

P_1 P_2 P_3

→ $P_n \approx C_n - 1n$ → path is circle without edge $1n$

Important graphs

empty graphs $n \geq 1$ $E_n([n], \emptyset)$ → E_n empty graphs with empty egde set

$V(E_n) = [n]$

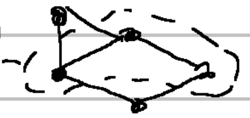
$E(E_n) =$ empty

example:

$E_1 \cong K_1 \cong P_1$ $E_2 \cong K_1 + K_1 = 2K_1$ E_3 E_4 ....

Induced Subgraphs

for a graph $G_n = (V, E)$, vertex Subset $A \subseteq V$ the induced subgraph G_A is defined as $V(G_A) = A$, $E(G_A) = \{uv \in E \mid u, v \in A\}$

example A  G ⇒ G_A: 

notation: $G_A \subseteq G$

note: A can be empty, but we usually don't talk about graphs without vertices

induced subgraphs can also fall into other graph-categories

Clique and independet set

Def: $G = (V, E)$ graph, $A \subseteq V$

→ A is a <u>clique</u>, if G_A is a complete graph

→ A is an <u>independent set</u>, if G_A is an empty graph (has no edges)

⇒ any single vertex is a clique AND an independend set

→ $\omega(G) =$ maximum size of A, so that $A \subseteq V(G)$ is a clique

G:  → $\omega(G) = 3$

→ $\omega(G) = \max\{|A| : A \subseteq V(G) \text{ is a clique}\}$

→ $\alpha(G) = \max\{|A| : A \subseteq V(G) \text{ is independent set (i-set)}\}$

→ as long as graph has at least one vertex: $\omega(G), \alpha(G) \geq 1$.

$\omega(G) =$ clique number $\alpha(G) =$ Independence number

Notation: disjoint union (of sets)

→ only allowed to be used for disjoint unions

$A + B = A \cup B$ but if $A \cap B = \emptyset$

Partition

Def.: partition of a set V is $V_1 + \dots + V_t = V$

⇒ partition of V in t parts, $t \geq 1$

⇒ every vertex is in exactly one part of a partition

Coloring and clique cover

Def.: $G = (V, E)$, $A \subseteq V$, $V_1 + \dots + V_t = V$

→ $V_1 + \dots + V_t$ is a coloring, if V_i is an i-set $\forall i \in [t]$

→ $V_1 + \dots + V_t$ is a clique cover, if V_i is a clique $\forall i \in [t]$

note: if $|V_i| = 1$ $\forall i \in [t]$, $V_1 + \dots + V_t$ is a coloring and a clique cover.

→ interesting question: find minimum t, so that $V_1 + \dots + V_t$ is a coloring or a clique cover

$\chi(G) = \min \{ t : \exists \text{ coloring } V_1 + \dots + V_t \text{ of } G \}$ → chromatic number

$\kappa(G) = \min \{ t : \exists \, V_1 + \dots + V_t \text{ clique cover of } G \}$ → clique cover number

$1 \leq \chi(G), \kappa(G) \leq |V|$

↑ ausgesprochen: Kappa

Numbers for important Graphs

	K_n	$K_{n,m}$	C_n	P_n	E_n	
$\omega(G)$	n	2	$\begin{cases} 3, & n = 3 \\ 2, & n \geq 4 \end{cases}$	$\begin{cases} 1, & n = 1 \\ 2, & n \geq 2 \end{cases}$	1	clique number
$\alpha(G)$	1	$\max(n,m)$	$\lfloor n/2 \rfloor$	$\lceil n/2 \rceil$	n	independence number
$\chi(G)$	n	2	$\begin{cases} 3, & n \text{ odd} \\ 2, & n \text{ even} \end{cases}$	$\begin{cases} 1, & n = 1 \\ 2, & n \geq 2 \end{cases}$	1	chromatic number
$\kappa(G)$	1	$\max(n,m)$	$\lceil n/2 \rceil$	$\lceil n/2 \rceil$	n	clique cover number

note: 2-colorable ⇔ $\chi(G) \leq 2$ ⇔ bipartite

$\lfloor n/2 \rfloor$ → n dived by 2 rounded down

$\lceil n/2 \rceil$ → n devided by 2 rounded up

	K_n	$K_{m,n}$	C_n, $n \geq 4$	P_n, $n \geq 2$	E_n
largest clique $\omega(G)$	n	2	2	2	1
smallest coloring $\chi(G)$	n	2	$n + (n \bmod 2)$	2	1
largest i-set $\alpha(G)$	1	$\max(m,n)$	$\lfloor \frac{n}{2} \rfloor$	$\lceil \frac{n}{2} \rceil$	n
smallest clique cover $k(G)$	1	$\max(m,n)$	$\lceil \frac{n}{2} \rceil$	$\lceil \frac{n}{2} \rceil$	n

clique number $\omega(G)$: size of a largest clique
chromatic number $\chi(G)$: smallest number of colors, that can be used to color G
independendce num. $\alpha(G)$: size of a largest independent set
smallest clique cover $k(G)$: smallest number of cliques necessary to cover the graph

<u>Observation</u>: For every graph, we have:
- $\chi(G) \geq \omega(G)$
- $k(G) \geq \alpha(G)$

<u>Proof:</u> IF $I \subseteq V_G$ is independent,
$C \subseteq V_G$ is a clique,
THEN: $|I \cap C| \leq 1$ → can be disjoint or can have a max of 1 vertex in common, due to definitions of I and C

hence for ANY coloring $V_1 + \dots + V_t = V_G$
and ANY clique C, we have $|C \cap V_i| \leq 1$, $i = 1, \dots, t$
IF $|C| = \omega(G)$, THEN $t \geq |C|$, THUS $\chi(G) \geq \omega(G)$

same for ANY clique cover $V_1 + \dots + V_t = V_G$
and ANY i-set I, we have $|I \cap V_i| \leq 1$, $i = 1, \dots, t$
IF $|I| = \alpha(G)$, THEN $t \geq |I|$, THUS $k(G) \geq \alpha(G)$

Main question of AGT: when does it hold, that:
1) $\chi(G) = \omega(G)$
2) $k(G) = \alpha(G)$

<u>boring answer</u>: any graph G with $\chi(G) > \omega(G)$: just add $K_n, n = \chi(G)$ to the graph, $G' = G + K_n$, $\omega(G') = n = \chi(G)$
$\chi(G') = n = \chi(G)$

K_n: complete graph with $|V| = n$

reformulated main question: When does P1) $\chi(G_A) = \omega(G_A)$ hold
P2) $k(G_A) = \alpha(G_A)$

for ALL induced subgraphs G_A for a graph G

$\Rightarrow$ (P1) $\forall A \subseteq V_G : \chi(G_A) = \omega(G_A)$

(P2) $\forall A \subseteq V_G : k(G_A) = \alpha(G_A)$

Graphs, that have this property: complete graphs K_n, $n \in \mathbb{N}$
empty graphs E_n, $n \in \mathbb{N}$
complete bipartite graphs $K_{m,n}$, $m,n \in \mathbb{N}$
paths P_n, $n \in \mathbb{N}$
even Cicles C_n, $n \in \mathbb{N}$, $n \bmod 2 = 0$
⊗

Observation: IF graphs G and H are vertex-disjoint
$\alpha(G+H) = \alpha(G) + \alpha(H)$
$k(G+H) = k(G) + k(H)$

⊗ for odd cicles: for $A \subset V_G$, $\chi(G_A) = \omega(G_A)$ and $k(G_A) = \alpha(G_A)$, but for $A = V_G$, those propeties do NOT hold, so odd cicles do not have properties (P1) and (P2)

Definition: graph G is called perfect, if G has properties (P1) and (P2)

smallest graph, that can break (P_1) and (P_2) is C_5. For graphs with $|V| \leq 4$, (P_1) AND (P_2) ALWAYS holds!
(even C_3 is odd, C_3 does not break (P_1) or (P_2) because $C_3 = K_3$).

Weak perfect graph theory (WPGT)

For every graph G, it holds, that:
G has $(P_1) \Leftrightarrow G$ has (P_2)

} only true, if considered for ALL induced subgraphs

Warning: $\forall A \subseteq V_G: \quad \chi(G_A) = \omega(G_A) \Leftrightarrow k(G_A) = \alpha(G_A)$
IS NOT TRUE !

→ (P_1) and (P_2) can break on different subsets of vertices, so there can be graphs with $\chi(G_A) = \omega(G_A)$ and $k(G_A) > \alpha(G_A)$

Complement

For a graph $G = (V, E)$, the complement of G is the graph $\overline{G} = (V, \overline{E})$, where $\overline{E} = \binom{V}{2} - E$ (all edges, exept those, that are in G)

example: E_3 is complement of K_3

graph G	$A \subseteq V$ is clique $\omega(G) = \alpha(\overline{G})$	Coloring $V_1 + \dots + V_k = V$ $\chi(G) = k(\overline{G})$
graph $\overline{G}$	A is i-set $\alpha(G) = \omega(\overline{G})$	clique cover $V_1 + \dots + V_k = V$ $k(G) = \chi(\overline{G})$

if G has property $(P1)$, $\overline{G}$ has property $(P2)$
if G has property $(P2)$, $\overline{G}$ has property $(P1)$
⇒ if G is perfect, $\overline{G}$ is perfect

Necessary tools to proof WPGT

To proof the WPGT, we consider a property (P3)

$$(P3)\ \forall A \subseteq V_G : \quad \omega(G_A) \cdot \alpha(G_A) \geq |A|$$

(for C_5, that does not hold, as $\alpha \cdot \omega = 2 \cdot 2 \ngeq 5 = |A| \rightarrow C_5$ not perfect, as we saw)

We will proof : G has $(P1) \Leftrightarrow G$ has $(P3)$

$\Updownarrow$

G has $(P2)$ $\Leftarrow$

Vertex replication / repition

Def.: For a grap $G = (V, E)$ and $\overbrace{h \in \mathbb{N}^V}^{\text{Vector}}$, $\mathbb{N} = \{0, 1, \dots\}$

we define $G \circ h$ as the graph on

- vertex - set $V(G \circ h) = \bigcup_{v \in V_G} \{v^1, \dots, v^{h(v)}\}$

- edges $u^i v^j$ if and only if $i \in [h(u)]$, $j \in [h(v)]$, $uv \in E_G$

called vertex replication / repition of G

note: h is a vector, that tells us, how many repetitions we want to have of the vertices of the original graph. if $h = \mathbb{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$, we get the original graph G, if $h = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$, we get no vertices and therefore no edges.

We will proove: if graph G has $(P1)$ / $(P2)$ every vertex repition of G has $(P1)$ / $(P2)$ $\Rightarrow$ if G perfect, every vertex - repetion of G perfect

Elementary operations

Def.: let $\mathbb{1}$ be $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{N}^V$, graph $G = (V, E)$;

let $i \in [|V|]$, $e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$ be the i-th unit vector in $\mathbb{N}^V$

vertex $v \in V$ define $G \circ v$ as $G \circ h$ with $h(x) = \begin{cases} 1 & \text{if } x \neq v \\ 2 & \text{if } x = v \end{cases}$

$h = \mathbb{1} + e_i$ if v is the i-th vertex $\rightarrow$ doubling vertex

and $G - v$ as $G \circ h$ with $h = \mathbb{1} - e_i$ $\rightarrow$ removing vertex

Observation: Every $G \circ h$ can be obtained from G by a sequence of elementary operations

More necessary tools

Lemma 2.6 : (numbering refers to lecture notes)
For G and $H = G \circ h$, we have : (i) G has $(P1) \Rightarrow H$ has $(P1)$
(ii) G has $(P2) \Rightarrow H$ has $(P2)$

Proof for Lemma 2.6:
(i) without loss of generality (wlog): $H = G \circ v$ or $H = G - v$

- case $H = G - v$: then H is an induced subgraph of G, so $H = G_{(V-v)}$ hence if G has $(P_1) \Rightarrow H$ has (P_1) because if G has (P_1) all induced subgraphs of G have (P_1)
- case $H = G \circ v$: $v \in V_G \sim v^1, v^2 \in V_H$
  $H - v^1 \cong G \cong H - v^2$
  take any $A \subseteq V_H$: if $|A \cap \{v^1, v^2\}| < 2$, then A is a subset of G and hence has $(P1)$
  so $\chi(H_A) = \chi(G_A) = \omega(G_A) = \omega(H_A)$
  let $v^1, v^2 \in A$. then: consider $A' = A - v^1 \subseteq V_G$
  because G has $(P1)$: $\chi(G_{A'}) = \omega(G_{A'})$
  v^1 has the same neighbors as v^2 due to definition
  so: $\chi(H_A) \leq \chi(G_{A'}) = \omega(G_{A'}) \leq \omega(H_A)$
  but as $\chi(G) \geq \omega(G)$ hold true for ANY graph G
  that means $\chi(H_A) = \chi(G_{A'}) = \omega(G_{A'}) = \omega(H_A)$

Continuing to proof WPGT

05.05.25

Proof for Lemma 2.6

(ii) • let G have (P2) ; wlog $H = G \circ x$ (doubling vertex x), let x, x' be the two copies of x in x. (So H is G with vertex x doubled)

- wlog A' contains x and x', let $A = A' - x' \subseteq V_G$
- (P2) for $G \rightarrow k(G_A) = \alpha(G_A) \Rightarrow V_1 + \dots + V_t$ clique cover of $G_A = H_A$ with $t = \alpha(H_A)$
- Every independent set I of H_A with $|I| = t$
  ↳ every i-set of H_A has to contain one vertex of every part in the clique cover

<u>Case 1</u>: $\exists\, I \subseteq A$ i-set of H_A, $|I| = t : x \in I$

then $I + x'$ is an i-set in $H_{A'}$

$\Rightarrow \alpha(H_{A'}) = t+1$

$\Rightarrow V_1 + \dots + V_t + \{x'\}$ is a clique cover of $H_{A'}$

$\Rightarrow k(H_A) \leq t+1 \leq \alpha(H_{A'})$

<u>case 2</u>: $\forall\, I \subseteq A$ i-set in H_A, $|I| = t : x \in I$

• let $C = G - x$, $\Rightarrow H_{A-C}$ has $\alpha(H_{A-C}) \leq t-1$

(P2) for $G \Rightarrow \exists$ clique cover of $G_{A-C} = H_{A-C}$ with $\leq t-1$ cliques $V_1' + \dots + V_{t-1}'$

$\rightsquigarrow V_1' + \dots + V_{t-1}' + C + x'$ clique cover of $H_{A'}$

$\Rightarrow k(H_{A'}) \leq t \leq \alpha(H_{A'}) \leq k(H_{A'})$

for all graphs

■ Lemma 2.6

Lemma 2.7

If $H = G \circ h$ then: (P2) for all induced subgraphs of G (but not necessarily for complete G) and (P3) for G $\Rightarrow$ (P3) for H

$\hookrightarrow \forall A \subseteq V_G,\ A \neq V_G,\ \kappa(G_A) = \alpha(G_A)$

Proof:
- Assume fsoc (for sake of contradiction $\Rightarrow$ Widerspruchsbeweis) that (P3) does not hold for H
- wlog $\forall A \subseteq V_G,\ A \neq V_G:\ \omega(H_A) \cdot \alpha(H_A) \geq |A|$

  let $\underset{w}{\omega(H)} \cdot \underset{a}{\alpha(H)} < |V_H|$

- some vertex s of G has $h(s) = h \geq 2$

$\rightsquigarrow$ in H $S = \{s_1, \dots, s_h\}$

- consider $H - s_h$ has (P3)

  $\Rightarrow |V_H| - 1 \leq \omega(H - s_h) \cdot \alpha(H - s_h) \leq \omega(H) \cdot \alpha(H) = w \cdot a \leq |V_H| - 1$

so $\Rightarrow |V_H| - 1 = \omega(H - s_h) \cdot \alpha(H - s_h) = \omega(H) \cdot \alpha(H) = w \cdot a = |V_H| - 1$

$\Rightarrow w \cdot a = |V_H| - 1,\ \alpha(H - s_h) = a,\ \omega(H - s_h) = w.$

$\Rightarrow \underbrace{\alpha(H - S)}$ $= a$ because s_h not important, so no copy of s crucial

because $\alpha(H) = a = \alpha(H - s_h)$

$\hookrightarrow$ otherwise we could use s_h for a largest i-set, but we can't, so...

- $H - S$ obtained from $G - s$ by vertex multiplication
- $G - s$ has (P2) because it's a true induce subgraph of G and by Lemma 2.6 $H - S$ has P2 [$G - s$ has (P2) $\xRightarrow{\text{Lemma 2.6}}$ $H - S$ has (P2)]

$\rightsquigarrow$ clique cover $V_1 + \dots + V_a$ of $H - S$

$|V_H - S| = |V_H| - h = w \cdot a - (h - 1)$

- $|S| = h \leq a = \alpha(H)$ since S i-set in H. at most $h - 1$ of $V_1 + \dots + V_a$ have size $< w$
- wlog $|V_1| = \dots = |V_{a-(h-1)}| = 1$
- let $X = V_1 + \dots + V_{a-(h-1)} + s_1$
- $\omega(H_X) = \omega(H) = w$
- since (P3) holds for H_X $\rightarrow$ $\alpha(H_X) = \left\lceil \frac{|X|}{\omega(H_X)} \right\rceil = \left\lceil \frac{w\,(a - (h-1)) + 1}{w} \right\rceil$

  $= \left\lceil a - (h-1) + \frac{1}{w} \right\rceil = a - (h-1) + 1$

$\therefore \exists\ I$ i-set in H_X, $|I| = a - (h-1) + 1$ $\quad s_1 \in I \rightsquigarrow I + \{s_1, \dots, s_h\}$ i-set in H

Proof of WPGT

- let $G=(V,E)$ be a graph. We proof $(P1) \Leftrightarrow (P2) \Leftrightarrow (P3)$ for G by induction. on $|V|$.

if $|V|=1$, then $\omega(G)=1$, $\chi(G)=1$, $\alpha(G)=1$, $\kappa(G)=1$, $\omega(G)\cdot\alpha(G)=1 \geq |V|$

to show $(P1) \Rightarrow (P3)$:
- say $(P1)$ holds for G
- let $A \subseteq V_G$
- if $A \neq V_G$, (P_1) holds for G_A

Induction Hypothesis IH: $\rightarrow$ $(P3)$ for G_A, i.e. $\omega(G_A)\cdot\alpha(G_A) \geq |A|$
- wlog $A=V_G$, i.e. we need to show $\omega(G)\cdot\alpha(G) \geq |V_G|$
- $(P1) \Rightarrow \exists$ coloring $V_1 + \dots + V_t = V_G$ with $t=\omega(G)$
- $|V_i| \leq \alpha(G)\ \forall i \Rightarrow \omega(G)\cdot\alpha(G) \geq |V_G|$

to show $(P3) \Rightarrow (P1)$:
- let $(P3)$ hold for G. To show $(P1)$ for G it is enougl (wlog) to show $\chi(G) \leq \omega(G)$
- consider all cliques of size $\omega(G)$

<u>Case 1</u>: $\exists I$ i-set in G: $\forall c$ clique, $|c|=\omega(G)$: $I \cap c \neq \emptyset$
- consider $G-I \rightarrow \omega(G-I) \leq \omega(G)-1$
- by IH, we have $(P1)$ for $G-I$: $V_1+\dots+V_t = V_G - I$ with $t \leq \omega(G)-1$

$\rightsquigarrow V_1+\dots+V_t+I$ coloring of $G \Rightarrow \chi(G) \leq t+1 \leq \omega(G)-1+1$, so $\chi(G) \leq \omega(G)$

<u>Case 2</u>: $\forall I$ i-set $\exists$ clique $C(I)$, $|C(I)|=\omega(G)$, $C(I)\cap I=\emptyset$

$\hookrightarrow$ for every i-set, there is a largest corresponding clique, that does not share a vertex with the indipendent set

Proof for WPGT - (P3) ⇒ (P1) continued 17.05.2025

case 2: $\cdot \forall I$ i-set $\exists$ clique $C(I)$, $|C(I)| = \omega(G)$, $C(I) \cap I = \emptyset$

- $Y = \{I \subseteq V_G \mid I \text{ is i-set}\}$
- $h(v) = |\{I \in Y \mid C(I) \ni v\}|$
- consider $H = G \circ h$
- (P3) for G → (P2) for G_A : $\forall A \subsetneq V_G$ $\Big\}$ $\xRightarrow{\text{Lemma 2.7}}$ (P3) for H
- $\omega(H) \cdot \alpha(H) \geq |V_H| = |X|$, say $X = V_H$ (I with vertex-duplication)
- $|X| = \sum_{v \in V_G} h(v) = \omega(G) \cdot |Y|$
- $\omega(H) \leq \omega(G)$
- $\alpha(H) = \max_{I \in Y} \{\sum_{v \in V_G} h(v)\} = \max_{I \in Y} \{\sum_{I' \in Y} \underbrace{|C(I') \cap I|}_{0 \text{ or } 1}\} \leq \underbrace{|Y| - 1}_{\text{because } C(I) \cap I = \emptyset}$
- $\omega(G) \cdot (|Y| - 1) \geq \omega(H) \cdot \alpha(H) \geq |X| = \omega(G) \cdot |Y|$ ↯

  ⇒ case 2 cannot happen!

(P2) ⇔ (P3)

- (P2) for G ⇔ (P1) for $\bar{G}$ ⇔ (P3) for $\bar{G}$ ⇔ (P3) for G

  □ WPGT

Equivalent:

- (P1) for G
- (P2) for G
- (P3) for G
- G perfect
- $\bar{G}$ perfect

non-perfect graphs:

- odd cycle C_t, $t \geq 5$
- complements of odd C_t, $t \geq 5$
- graphs with induced odd C_t, $\bar{C}_t$, $t \geq 5$

⇒ these are all non-perfect graphs

Strong Perfect Graph Theorem (SPGT)

For every graph G it is equivalent:

- $C_t, \overline{C_t}$ for $t \geq 5$ odd is no induced subgraph of G
- G is perfect

!

Intersection representation

Def.: $\mathcal{S} = \{S(v) \mid v \in V\}$ collection of sets

is an intersection representation of $G = (V, E)$ if:

$uv \in E \Leftrightarrow S(u) \cap S(v) \neq \emptyset.$

Intervalgraph

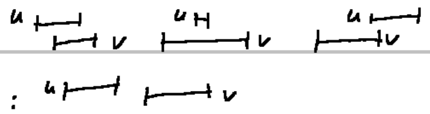
Def.: G is an intervalgraph, if G has an intersection representation with intervals of $\mathbb{R}$, i.e. $\mathcal{I} = \{I(v) \mid v \in V\}$

$\forall: I(v) = [l_v, r_v] \subseteq \mathbb{R}$

$uv \in E \Leftrightarrow I(u) \cap I(v) \neq \emptyset \Leftrightarrow \min\{r_u, r_v\} \geq \max\{l_u, l_v\}$

Bsp: $uv \in E$:

$uv \notin E$:



t-holes & t-antiholes

Def.: For a graph G, integer $t \geq 4$

- a t-hole in G is an induced subgraph $G_A \cong C_t$
- a t-antihole in G is an induced subgraph $G_A \cong \overline{C_t}$

(SPGT: G perfect $\leftrightarrow$ no odd t-hole / t-antihole)

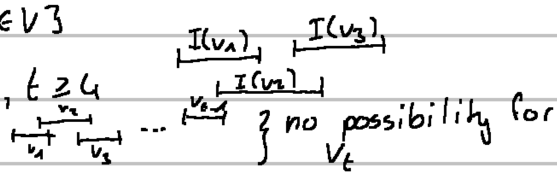
Lemma: G interval graph $\Leftrightarrow$ no t-hole, $t \geq 4$

Proof: interval representation $\mathcal{I} = \{I(v) = [l_v, r_v] \mid v \in V\}$

- Assume, there is a t-hole $C_t = [v_1, v_2, \dots, v_t]$, $t \geq 4$

$\rightarrow I(v_1), I(v_3)$ cover distinct endpoint of $I(v_2)$

$\Rightarrow I(v_1) \cap I(v_2) = \emptyset \Rightarrow v_1 v_2 \notin E$ ↯



Lemma: G intervalgraph $\Rightarrow$ G perfect

Proof: We use SPGT

- no odd t-hole, see previous Lemma
- no odd t-antihole, see previous Lemma $\qquad \overline{C_5} = C_5$
- 4-hole in $\overline{C_t}$ for $t \geq 7$ does not happen by prev. Lemma $\square$

We showed:

G is intervalgraph $\Rightarrow$ G has no odd t-(anti-) holes $\Rightarrow$ G is perfect.

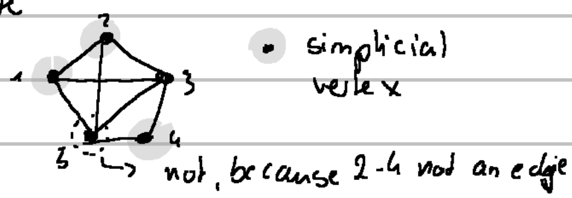
interval graph → no t-hole >4 → no odd hole ⟺ perfect
no odd antihole SPGT

③ Chordal Graphs

<u>Def:</u> $G=(V,E)$ is chordal if G has n t-hole, $t>4$.
Equally, every cicle C_t, $t\geq 4$ in G has a chord,
i.e. an edge uv with uv not adjacent on cycle
↳ uv not neighbours on cycle

⇒ induced cycles have max 3 vertices, cycles with 4 or more vertices must have (a) chord(s)
<u>examples:</u> complete graphs, interval graphs, paths, empty graphs, $\boxed{\text{trees}}$
trees have leaves
devide & conquer

<u>Def.:</u> $G=(V,E)$ graph, vertex $v\in V$ is simplicial if $\underbrace{Adj(v)}=\{u\in V \mid uv\in E\}$ is a clique. (neighbours also all pairwise adjacent) vertices adjecant to v
↳ neighbours
(In a tree, only leaves are simplicial. In complete graphs, all vertices are simplicial)



<u>Goal:</u> Show that every chordal G has ≥ 1 simplicial vertices.
<u>Lemma 3.6.</u> every chordal G has one or more simplicial vertices.

<u>Lemma:</u> $\left.\begin{array}{l} v \text{ simplicial in } G \\ G-v \text{ perfect}\end{array}\right\}$ G is perfect. (adding simplicial node retains graph perfection)

<u>Proof:</u> Verify (P1): $\forall A\subseteq V$: $\chi(G_A)=\omega(G_A)$

- consider only fixed $A\subseteq V_G$
  ↳ case: $v\notin A$, then $A\subseteq V_{G-v}$ and $\chi(G_A)=\omega(G_A)$, as $G-v$ is perfect (has (P1)).
  ↳ case 1: $v\in A$: let $A'=A-v\subseteq V_{G-v} \Longrightarrow \chi(G_{A'})=\omega(G_{A'})$
  → coloring $A'=V_1+\dots+V_t$, $t=\omega(G_{A'})$
  → $\exists i$ (es existiert ne Farbe für G, sodass die kein Nachbar von v h ⇒ v kann diese Farbe annehmen ⇒ alles gut)

↳ case 2: $Adj(v) \cap A' \geq t = \omega(G_{A'})$

$\rightsquigarrow$ is a clique in $G_{A'}$ $\Rightarrow$ $|Adj(v) \cap A'| = \omega(G_{A'}) = t$

$\rightsquigarrow$ $(Adj(v) \cap A') \cup \{v\}$ is a clique in G_A of size $t+1$

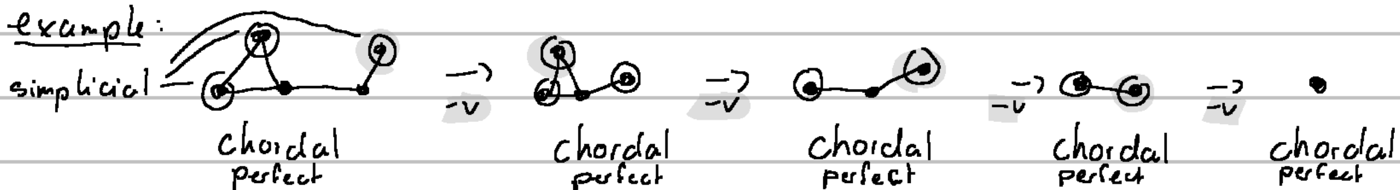
$\rightsquigarrow$ $\omega(G_A) \geq t+1 = \omega(G_{A'})+1 = \chi(G_{A'})+1 \geq \chi(G_A) \geq \omega(G_A)$

$\rightsquigarrow$ coloring $V_1 + \ldots + V_t + \{v\}$ on $t+1$ colors $\square$.

<u>Observation:</u> $G = (V,E)$ is chordal $\Rightarrow$ $\forall A \subseteq V_G$ G_A is chordal, in particular $G-v$ is chordal $\forall v \in V$

↳ removing vertices does NOT destroy chordality

<u>example</u>:



by removing vertices one at a time, one eventually arrives at K_1

(we do: interval graphs are chordal + chordal graphs are perfect

$\Rightarrow$ interval graphs are perfect ; without SPGT)

<u>Def.:</u> $G = (V,E)$, $|V| = n$.

A <u>perfect elimination scheme</u> (PES) of G is a vertex coloring

$\sigma = [v_1, \ldots, v_n] > t$, v is simplicial in $G_{[v_1, \ldots, v_n]}$

(eine Reihenfolge, in der man Knoten entfernt. Die entfernten Knoten müssen zum Zeitpunkt des Entfernens simplicial sein. Knoten, die nicht simplicial sind können durch das Entfernen von Nachbarknoten simplicial werden. Siehe example oben. Für complete Graphs jede Reihenfolge, für Pfade immer ein Knoten am Ende, für Bäume immer Blätter entfernen ist PES).

previous Lemma $\Rightarrow$ graphs with PES's are perfect.

Notation, convention for vertex orderings: $\sigma = [v_1, \ldots, v_n]$

if you lign up the vertices sorted from v_1 to v_n, then the neighborhood on the right of the respective vertex is a clique

← In every PES, every right neighbourhood is a clique

Def.: $G=(V,E)$ graph, $S \subseteq V$ is a separator, if $G-S$ is disconnected.
(disconnected: G has at least 2 components, that are not connected by an edge)
↳ example: △ ▽

S is an ab-seperator, if a,b not adjacent in G and a,b in different components of $G-S$.
[If graph disconnected already, empty set is seperator]

Goal: find a seperator, that is a clique in a chordal graph that is NOT complete.
tree: every single vertex, that is not a leave is a seperator and a clique
complete Graphs: have no seperators

Lemma 3.4: if G is chordal, a and b distinct vertices, $ab \notin E_G$. $\}$ $S \subseteq V_G$ inclusion-minimal ab-seperator $\} ====$
$===\!\!>$ S is a clique.

(inclusion minimal: we use the minimal amount of vertices, that are an ab-seperator.
↳ every true subset of S is no longer an ab-seperator. If G already disconected, a,b in different components $\Rightarrow$ $S = \{\}$. → empty set).
↳ There might be several inclusion-minimal ab-seperators. They can have different sizes. important criterion is, that one cannot remove vertices from it, so that a and b are still disconnected.

Proof: - If $|S| \leq 1$, then S is a clique. (S inclusion-minimal ab-seperator)
• so $|S| \geq 2$, take $x, y \in S$, $x \neq y$. We show, that $xy \in E$.
- $S-x$ is NOT an ab-seperator
- G_A, G_B are the components of $G-S$ with $a \in G_A$, $b \in G_B$
↳ x has edge to A and to B. so does y.
↳ we have a cycle $C = [x, a_1, \dots, y, \dots, b]$ (and back to x).
Choose C, so that it is a shortest such cycle.
- C has at least 4 vertices (a, b, x, y).
- G is chordal $\Rightarrow$ C has chord e.

Where is e? → not between a_i, a_j / b_i, b_j becaus C is shortest, not between a_i and b_i because a,b not adjacent $\Rightarrow$ between x and y

Where is e?

not between a_i, a_j / b_i, b_j because C is shortest cycle

not between a_i, b_i becaus G_A and G_B distinct components

not between x, a_i / x, b_i / y, a_i / y, b_i because C shortest

$\Rightarrow$ e is between x and y

Lemma 3.4 19.05.25

G chordal, $a, b \in V_G$, $ab \notin E_G$, S inclusion minimal ab-separator
$\Rightarrow$ S clique

Lemma 3.6

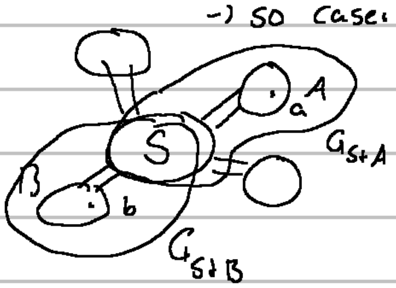
G chordal $\Rightarrow$ G has at least on simplicial vertex
$\Rightarrow$ If $G \neq K_n$ then G has at least two non-adjacent simplicial vertices

Proof: by induction of $n = |V_G|$

$n = 1 \to G = K_1$ ✓

$n \geq 2 \to$ case: $G = K_n \to$ every vertex is simplicial ✓

$\to$ so case: $G \neq K_n \to$ let $a, b \in V_G$, $ab \notin E_G$



$\to$ let S be an inclusion minimal ab-separator

$\to$ component G_A contains a, component G_B contains b
↳ of $G - S$

$\to$ apply induction on G_{S+A} and G_{S+B}

$\rightsquigarrow$ in G_{S+A}: either all vertices simplicial, then complete graph
or there are two non-adjacent simplicial vertices

$\Rightarrow \exists$ simplicial vertex $x \in A \rightsquigarrow$ simplicial in G_{S+A} but also in G

- symmetric: $\exists\, y \in B$ simplicial in G
- $xy \notin E_G$

G chordal $\longrightarrow$ $G - v_1$ chordal, perfekt

v_1 simplicial, perfekt $\longleftarrow$ v_2 simplicial

$\downarrow \uparrow$

PES

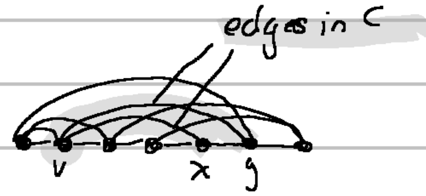
$G = K_1$ $\longleftarrow$ chordal

perfekt v_n $\longrightarrow$ perfekt

(i) G chordal, i.e. every cycle with length ≥ 4 has a chord
(ii) every induced cycle is a triangle (not a ℓ-hole)
$\Downarrow$ Lemma 3.4
(iii) every inclusion minimal seperator is a clique
$\Downarrow$ Lemma 3.6
(iv) every induced subgraph has at least one simplicial vertex
$\Downarrow$
(v) there is a PES (perfect elimination scheme)

⊛

<u>Proof of ⊛:</u>



- we have graph G, PES σ, cycle C with length ≥ 4

$\rightarrow$ v is leftmost vertex of C in σ, say $v = \sigma(i)$
- $x, y \in Adj(v) \cap \{\sigma(i+1), \dots, \sigma(n)\}$

$\overset{PES}{\Rightarrow}$ $xy \in E_G \rightarrow$ chord in C ◻

Testing $\overset{\text{naive}}{\text{time}}$: for (i) exponential (there could be exponential many cycles)
for (ii) exponential
for (iii) unclear, polynomial to find one, but all?
for (iv) exponential
for (v) exponential to go over all vertex orderings
↳ idea: find one simplicial vertex remove, find new simplicial vertex, ...
⇝ leads to recognition algorithm with runtime:

$$O(n \cdot \text{time\_to\_find\_simplicial\_vertex}) = O(n^4) \rightarrow \text{polynomial}$$

↳ either we get PES or we know that graph is not chordal.

next step: bring runtime down from polynomial to quadratic or even linear
$\Rightarrow$ first algorithm: algorithm tests, if there is a PES or graph is not chordal. PES can then be used in later algorithms

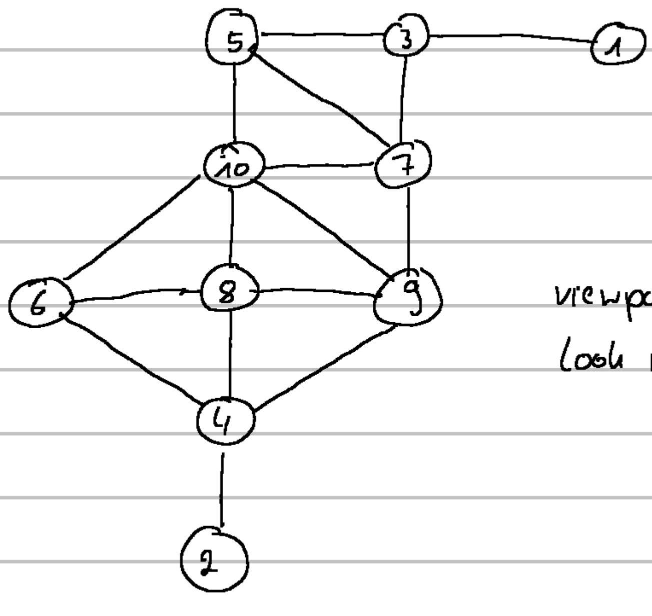
Algorithm 1: Lex BFS → Lexicographical breadth first search

input: undirected graph $G = (V, E)$

output: vertex ordering σ (not necessarily a PES yet)

⇒ first part of recognition algorithm.

1) assign each vertex label $\emptyset$; → empty string → everyone gets empty lable
2) for $i \leftarrow n$ to 1 do → first one random, second random neighbour of first
3) Choose a vertex v
   with no assigned number in σ
   with lexicographically largest label; } start with largest → right most vertex and move to smallest → leftmost
4) $\sigma(i) \leftarrow v$;
5) for every vertex $w \in Adj(v)$
   with no assigned number in σ
6) append i to label(w); → lable is string, letter i appended at end
7) end for
8) end for



viewpoint 1,
look next slide

Label of:	
10 :	$\emptyset$
9 :	10
8 :	10 . 9
7 :	10 . 9
6 :	10 . 8
5 :	10 . 7
4 :	9 . 8 . 6
3 :	7 . 5
2 :	4
1 :	3

⇒ algorithm is BFS with lables as tie-breakers. However, there are still ties left, that need to be broken, e.g. for the very first vertex

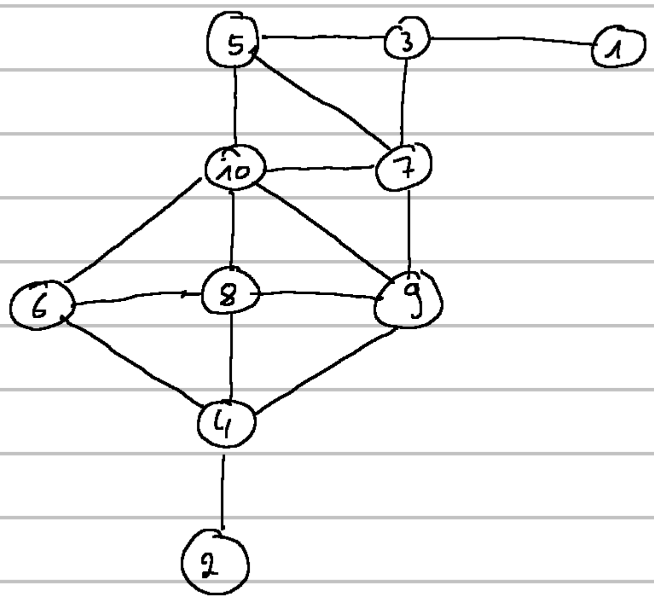
⇒ algorithm computes a vertex-ordering for every single undirected graph

Lex BFS:

viewpoint 1:
- lables at vertices
- strings over the alphabet $\{1, \dots, n\}$
- lexicographic: $1 < \dots < n$
- $\text{lable}(v) = \alpha_1 \dots \alpha_s$
- $\text{lable}(u) = \beta_1 \dots \beta_t$
- $\alpha = \alpha_1 \dots \alpha_s < \beta_1 \dots \beta_t = \beta$ $\begin{cases} \alpha_1 <_{lex} \beta_1 & \text{or} \\ \alpha = \emptyset & \text{or} \\ \alpha_1 = \beta_1 \text{ and } \alpha_2 \dots \alpha_s <_{lex} \beta_2 \dots \beta_t \end{cases}$

viewpoint 2:
- Q queue of all not-numbered vertices
- $v \leftarrow \text{first}(Q)$ 

- elements of Q are sets of vertices of the same lable sorted lexicographically in Q
  - ↳ sets get split up, when symbols are added to the lables and keeps the subsets in lexicographical order



$Q_0 = [\emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset]$

$Q_1 = [10\ 10\ 10\ 10 \mid \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset]$

$Q_2 = [10.9\ 10.9 \mid 10\ 10 \mid 9 \mid \emptyset\ \emptyset\ \emptyset]$

$Q_3 = [10.9 \mid 10.8 \mid 10 \mid 9.8 \mid \emptyset\ \emptyset\ \emptyset]$

$Q_4 = [10.8 \mid 10.7 \mid 9.8 \mid 7 \mid \emptyset\ \emptyset]$

$Q_5 = [10.7 \mid 9.8.6 \mid 7 \mid \emptyset\ \emptyset]$

$Q_6 = [9.8.6 \mid 7.5 \mid \emptyset\ \emptyset]$

$Q_7 = [7.5 \mid 4 \mid \emptyset]$

$Q_8 = [4 \mid 3]$

$Q_9 = [3]$

$Q_{10} = \emptyset$

Proof LexBFS: look at lecture slides

vertex ordering σ is PES if and only if
$\boxed{\text{slide}}$

vertex ordering is LexBFS-result, if and only if

Plan: given G:

- run LexBFS $\rightsquigarrow$ obtain vertex ordering σ

proof that
σ PES $\Leftrightarrow$ G chordal
↳ interesting direction: $\Leftarrow$

test if σ is PES in linear time

implement LexBFS in linear time

recognition algorithm

- $\sigma \leftarrow$ LexBFS(G)

↳ test σ for PES
  - $\Rightarrow$ YES $\rightarrow$ G is chordal
  - $\Rightarrow$ NO $\rightarrow$ G is not chordal

Lemma:

$\sigma \leftarrow$ LexBFS(G)

$\rightarrow$ then $\forall a,b,c \in V_G$ it holds:

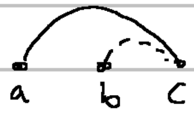
$$\left.\begin{array}{l} a <_\sigma b <_\sigma c \\ ac \in E_G,\ bc \notin E_G \end{array}\right\} \Rightarrow \exists d \in V_G : c <_\sigma d,\ ad \notin E_G,\ bd \in E_G$$

Lemma: 28.05.25

$\sigma \leftarrow LexBFS(G)$

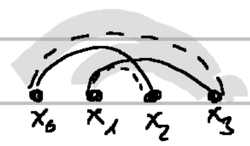
$\rightarrow$ then $\forall a, b, c \in V_G$ it holds:

$$\left.\begin{matrix} a <_\sigma b <_\sigma c \\ ac \in E_G,\ bc \notin E_G \end{matrix}\right\} \Rightarrow \exists d \in V_G : c <_\sigma d,\ ad \notin E_G,\ bd \in E_G$$

Proof: consider  — edge --- non-edge

- when c is processed by LexBFS.
  1) if $lable(a) = lable(b)$
     then afterwards $lable(a) >_{lex} lable(b)$ and thus will still hold, when b is processed, contradicts choice of b ↯

$\Rightarrow$ 2) $lable(a) \neq lable(b)$ then $lable(b) >_{lex} lable(a)$
  - consider step before first time, when $lable(a) \neq lable(b)$
  - when processing vertex d, $c <_\sigma d$
  - it holds $b \in Adj($

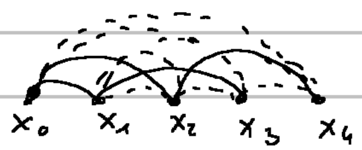
note:
if I have , I can conclude, that 

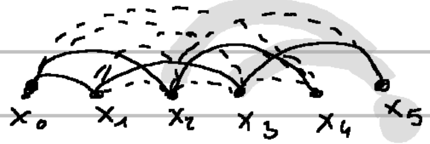
Theorem 3.9

G is chordal if and only if LexBFS outputs PES.

Proof: "$\Leftarrow$" clear (LexBFS outputs PES $\rightarrow$ G chordal)

"$\Rightarrow$": proove the contraposition, i.e. σ (output of LexBFS) not PES $\Rightarrow$ G not chordal
- σ not PES, then $\exists a, b, c$; $a <_\sigma b <_\sigma c$; $ab, ac \in E_G$; $bc \notin E_G$ $\quad a = x_0,\ b = x_1,\ c = x_2$
- by L3 (prev. Lemma): $\exists x_3 : x_2 <_\sigma x_3$; $x_1 x_3 \in E_G$; $x_0 x_3 \notin E_G$
  i) $x_2 x_3 \in E_G$: $\rightarrow x_0 x_1 x_2 x_3$ induce $C_4 \rightarrow G$ not chordal ✓
  ii) $x_2 x_3 \notin E_G$: by L3 $\exists x_4 : x_2 x_4 \in E_G,\ x_1 x_4 \notin E_G,\ x_3 <_\sigma x_4$
     - if $x_0 x_4 \in E_G$ contradicts the choice of x_2 as rightmost
     - so $x_0 x_4 \notin E_G$: 1) $x_3 x_4 \in E_G$: then $G[x_0, \dots, x_4] = C_5$ ✓
       2) $x_3 x_4 \notin E_G$: then $G[x_0, \dots, x_4] = P_5$ with endpoints x_3, x_4





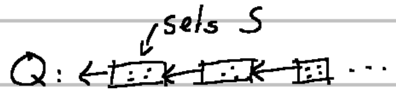
by L3 $\exists x_5$

- if $x_0 x_5 \in E_G$: then $x_0 x_2 x_5$ bad PES-triple with x_5 further to the right, contradiction ↯
- so $x_0 x_5 \notin E_G$
- if $x_1 x_5 \in E_G$ then ↯ choice of x_3
- so $x_1 x_5 \notin E_G$

1) $x_4 x_5 \in E_G \Rightarrow$ induced C_6 on $x_0 \dots x_5$

2) $x_4 x_5 \notin E_G \Rightarrow$ induced P_6 and argument continues.

LexBFS in $O(|V|+|E|)$



we want to split the sets by using the lables

Datastructure: Q: queue with sets
- First(Q)
- double linked list

S: set of vertices
- list but non-empty (if empty → remove)
- Flag(S) boolean, if S has been split

W: vertex
- S(w): set S, that includes w

Fix List L: list of sets, that have been split
- First(L)

Pseudocode on slides: ⇒ Algorithm 2

Analysis: line 1-9: linear in $|Adj(v)| \to O(|Adj(v)|)$

line 10-16: linear in $|FixList| \to O(|FixList|) = O(|Adj(v)|)$

Update step in $O(|Adj(v)| + \underbrace{|FixList|}_{\le |Adj(v)|})$

LexBFS total runtime: $O(\underbrace{\sum_v |Adj(v)|}_{= 2|E|} + |V|) = O(|V|+|E|)$

we still need to: test, if output of LexBFS is PES

Test, whether given vertex-ordering σ is a PES:

naive-aproach: •) test all triples for → $\Theta(n^3)$

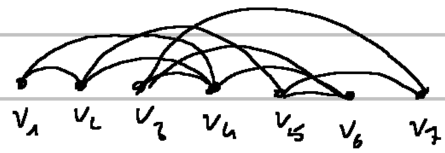
•) test right neighbour of each vertex → for clique $\sum_v |Adj(v)|^2 \approx O(n^3)$

↳ looks at vertices more than once → potential for improvement

=> Idea:

v tells it's leftmost right neighbour u a set of vertices, that should be pairwise adjacent (so they should form a clique) → v also wants u to be adjacent to all of those neighbours

→ Pseudocode on slides: Algorithm 3



} false: $\underbrace{v_5 \text{ in } A(v_4)}$, but not adjacent to v_4

↳ put there by v_1

for v_1: $X = \{v_2, v_4\}$, $u = v_2$; $A(u) \leftarrow$ add $X - u$: $A(v_2) = \{v_4\}$, $A(v_4) = \{v_5\}$

for v_2: $X = \{v_4, v_5\}$, $u = v_4$;

for v_4:

Theorem: 02.06.25

Algorithm 3 (lecture 8 - test for PES) is correct, that means:

Alg 3. returns true $\iff$ σ is PES for G

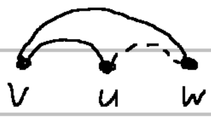
Proof:

Alg. 3 returns true $\iff$ σ is PES for G.

↳ equivalent: Alg. 3 returns false $\iff$ σ is NOT PES for G.

"$\Rightarrow$": $\exists u$ vertex with: $A(u) - Adj(u) \neq \emptyset$, say $w \in A(u) - Adj(u)$

↳ who put $w \in A(u)$? $\rightarrow$ some v earlier $\Rightarrow$ u leftmost in X_v, $w \in X_v - \{u\}$

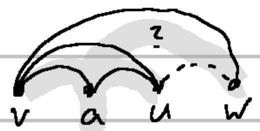
$\Rightarrow$  $\Rightarrow$ G not PES, pattern forbidden for PES

— edge --- non-edge

"$\Leftarrow$" assume σ not PES for G. then:

take  with v, u closest together

- claim: u is the leftmost right neighbour of v
  - if $a \in X_v$, $v < a < u$
    - $au \notin E_G$ $\rightsquigarrow$ ↯ contradicts choice of triple
    - $aw \notin E_G$ $\rightsquigarrow$ ↯ vaw better triple
    - $aw \in E_G$ $\rightsquigarrow$ ↯ auw better triple

 } all cases contradicted

$\Rightarrow$ so u is leftmost in X_v

$\Rightarrow$ Algo 3 put w into $A(u)$ when processing v

$\Rightarrow$ later (processing u) we have $w \in A(u) - Adj(u)$

$\Rightarrow$ Algo 3 returns FALSE $\square$

Theorem:

Algorithm 3 (lecture 8 - test for PES) can be done in $O(|V| + |E|)$

Proof: · for loop over each vertex v once

- Lines 2-7 possible in $O(|Adj(v)|)$
  ↳ Line 7: append $X - \{u\}$ to $A(u)$ without checking for duplicates

total $O(\sum_v |Adj(v)|) = O(|V| + |E|)$

- check line 8-10:
  $A(v) - Adj(v) \neq \emptyset$? in $O(|A(v)|)$

total $O(\sum_v |A(v)|) = O(\sum_v |Adj(v)|) =$
$= O(|V| + |E|)$

↳ strictly speaking in $O(|A(v)| + |Adj(v)|) = O(|A(v)|)$

⇒ we can recognize in linear time weather G is chordal and compute a PES σ of G.

We compute $\chi(G)$, $\omega(G)$, $\alpha(G)$, $k(G)$ for chordal graphs

⊗
- ▷ Algorithm 5 computes $\omega(G)$ and $\chi(G)$ (which are the same for chordal) with clique C, coloring Φ being optimal
- ▷ Algorithm 6 computes $\alpha(G)$ and $k(G)$ (which are the same for chordal graphs) with i-set U, cliquecover Ψ being optimal

⊗ lecture 8, algorithms 5 and 6

for notes, look at slides of lecture 8

Theorem:

Algorithm 5 (lecture 8) computes a clique C and a coloring Φ with $|C| = \omega(G)$ and $\max_v \Phi(v) = \chi(G)$

Proof: wlog: without loss of generality

1) C is a clique: C is of the form $\{v\} + X_v$.

↳ as G is PES: X_v is a clique

$\rightsquigarrow C = X_v + \{v\}$ is a clique

$\Rightarrow \max_v (|X_v| + 1) = |C| \leq \omega(G)$ □

2) Φ is a coloring: - we set $\Phi(v)$ once and never change, $\Phi(v) \geq 1$.

- let $uv \in E_G$: wlog $u \in X_v$

$\Rightarrow$ we choose $\Phi(v)$ to be different from $\Phi(u)$

$\Rightarrow$ valid coloring, because we never change $\Phi(u)$

$\Rightarrow \chi(G) \leq \max_v \Phi(v)$ □

3) C and Φ are both optimal:

- for every v: $\Phi(v) \leq |X_v| + 1$
- therefore $\max_v \Phi(v) \leq \max_v (|X_v| + 1) = |C|$

for every graph

$\Rightarrow \chi(G) \leq \max_v \Phi(v) \leq \max_v (|X_v| + 1) = |C| \leq \omega(G) \leq \chi(G)$

$\Rightarrow$ equality everywhere $\rightsquigarrow |C| = \omega(G)$, $\max_v \Phi(v) = \chi(G)$,

$\omega(G) = \chi(G)$ □

Theorem:

Algorithm 5 can be done in $O(|V| + |E|)$

Proof: - the for-loop iteration for vertex v takes $O(|Adj(v)|)$

- line 6 simmilar to Alg. 4 in $O(|X_v|) \to O(|V| + |E|)$ □

↳ go over all neighbours, cross out all taken numbers up to size of neighbourhood, then return min of the remaining.

Theorem:

Algorithm 6 computes i-set U and clique cover Ψ with $|U| = \alpha(G)$ and $\max_v \Psi(v) = k(G)$

Proof:

1) U is an i-set: invariant: $w \in U$, $v >_G w$, $\Psi(v) = 0 \Rightarrow vw \notin E_G$

- equivalent: $w \in U$, $v >_G w$, $vw \in E_G \Rightarrow \Psi(v) \neq 0$

true since $v \in X_w$ gets assigned $\Psi(v) \leftarrow |U| \neq 0$

$\Rightarrow |U| \leq \alpha(G)$ □

2) Ψ is a clique cover: set $\Psi(w) \leftarrow |U| =: i \;\; \forall w \in \underbrace{\{v\} + X_v}_{\text{clique since } G \text{ is PES}}$ (line 8)

never use this number again, only larger numbers. Cliques only shrink, but that is ok.

- never $|U| =: i$ assigned again.

$\rightarrow$ final Ψ : $\{w : \Psi(w) = i\} \subseteq \{v\} + X_v \rightarrow$ clique

$k(G) \leq \max_v \Psi(v) = |U| \leftarrow$ final size of U

3) U and Ψ are both optimal:

$k(G) \leq \max_v \Psi(v) = |U| \leq \alpha(G) \leq k(G)$ for every graph

$\rightarrow$ equality everywhere, i.e. $|U| = \alpha(G)$, $\max_v \Psi(v) = k(G)$,

$\alpha(G) = k(G)$ □

Theorem:

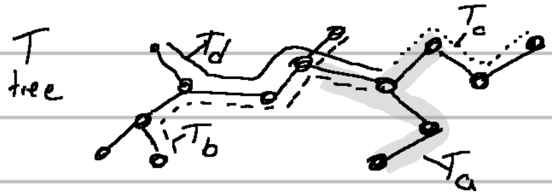
Algorithm 6 can be done in $O(|V| + |E|)$

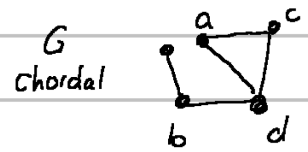
Proof: - simmilar to proof for algorithm 5.

23.08.25

- underlying tree $T = (V_T, E_T)$ of graph $G = (V, E)$
- vertex v of G $\rightsquigarrow$ subtree T_v of T
- edge uv of G $\rightsquigarrow$ intersecting (non-disjoint) subtrees

$$uv \in E_G \Leftrightarrow T_u \cap T_v \neq \emptyset .$$





Obs.: $\{T_a, T_b, T_c, T_d\}$ has Helly property

Plan:

chordal Graphs	=	intersection graphs of subtrees of a tree
$\supseteq$		$\supseteq$
interval Graphs	=	intersection graphs of subtrees of a path

main ingredient: Helly property

<u>Def.:</u> a family $\{A_i\}_{i \in I}$ of sets has the Helly-property if:
$$\forall J \subseteq I : (A_j \cap A_{j'} = \emptyset \quad \forall j, j' \in J) \Rightarrow \bigcap_{j \in J} A_j \neq \emptyset .$$

<u>Proposition 3.13</u>

T tree $\Rightarrow$ $\{T_i \subseteq T \mid T_i \text{ subtree}\}_{i \in I}$ has the Helly-property

<u>Theorem 3.14</u>

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

i) G is chordal

ii) $\exists$ tree $T = (V_T, E_T)$, $\{T_v \subseteq T \mid v \in V, T_v \text{ subtree}\}$, such that $vw \in E \Leftrightarrow T_v \cap T_w \neq \emptyset$

iii) $\exists$ tree $T = (V_T, E_T)$ such that
- $V_T = \{X \subseteq V \mid X \text{ is inclusion-maximal clique in } G\}$ and
- $\forall v \in V : K_v = \{X \in V_T \mid v \in X\}$ induces a subtree

Proof: next page

Proof for 3.14

(ii) ⇒ (i) : - let G be an intersection graph of subtrees of tree T

- let $C = [v_1, \dots, v_k]$, $k \geq 4$ cycle of G
  - $T_1 := T_{v_1} \cup T_{v_2}$ is subtree of T
  - $T_2 := T_{v_3} \cup \dots \cup T_{v_{k-1}}$ — " —
  - $T_3 := T_{v_k} \cup \dots \cup T_{v_k}$ — " —

Note: $T_1 \cap T_2 \neq \emptyset$ since $v_2 v_3 \in E_G$; $T_2 \cap T_3 \neq \emptyset$ since $v_3 v_4 \in E_G$;
$T_1 \cap T_3 \neq \emptyset$ since $v_1 v_k \in E_G$

$\xRightarrow{\text{Helly}}$ $\exists x \in V_T : x \in T_1 , x \in T_2 , x \in T_3$

Case 1: $x \in T_{v_1}$

→ then x adjacent to v_j , $j \in \{3, \dots, k-1\}$ with $x \in T_{v_j} \subseteq T_2$ → chord at $x v_j$

Case 2: $x \in T_{v_2}$

→ then x adjacent to v_j , $j \in \{4, \dots, k\}$ with $x \in T_{v_j} \subseteq T_3$ → chord at $x v_j$

(i) ⇒ (iii) : - let $G = (V, E)$ be chordal ; $K(G) = \{x \subseteq V_G : x \text{ inclusion-maximal clique in } G\}$

- we find tree T by induction on $|V_G|$

$\forall v \in V : K_v = \{x \in K(G) : v \in x\} \subseteq K(G)$

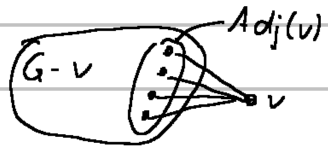
(*) induces subtree in T

base case: $|V_G| = 1$ → $V_T = K(G) = 1$ vertex

G :	$\overset{v}{\bullet}$
T :	$\bullet \{v\}$

step : $|V_G| \geq 2$: - let v be simplicial in G

- induction on $G - v$
  ↳ get T' tree of $K(G-v)$



case 1: $Adj(v) \in K(G-v)$

→ $Adj(v) + \{v\} \in K(G)$ and $K(G-v) - Adj(v) = K(G) - (Adj(v) + \{v\})$

relable $Adj(v)$ in T' to $Adj(v) + \{v\}$ in T

observe, that (*) still holds as $\forall w \neq v$, K_w did not change

$K_v = \{Adj(v) + \{v\}\}$

case 2: $Adj(v) \notin K(G-v)$

- let $x \in K(G-v)$, $Adj(v) \subseteq X$
- in T' there is a vertex for X
- add new vertex $Adj(v) + \{v\}$ adjacent only to x
- (*) still holds, as $\forall w \in Adj(v) : w \in X$

(iii) $\Rightarrow$ (ii) : Let $T := (V_T, E_T)$ be a tree with $(*)$

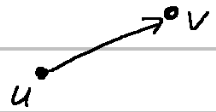
- take T_v = subtree induced by K_v in T
- $vw \in E_G \Leftrightarrow \exists X \in K(G) : \{v, w\} \subseteq X$

$\Leftrightarrow \exists X \in K(G) : X \in K_v,\ X \in K_w$

$\Leftrightarrow \exists X \in V_T : X = T_v \cap T_w$

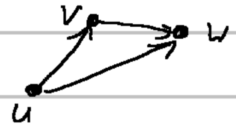
$\Leftrightarrow T_v \cap T_w \neq \emptyset$ $\blacksquare$

Chapter 4: Compatibility graphs



- vertices: Elements
- edges: „better-than" relations
  ↳ directed edges (u,v): „v is better than u"
- binary relation $\prec \subseteq V_G \times V_G$
  ↳ irreflective: $v \nprec v \quad \forall v \in V_G$
  ↳ transitive: $\forall u,v,w: u \prec v \land v \prec w \Rightarrow u \prec w$

  called „strict partial order"



Notation for this chapter:

25.06.25

- only directed edges
- graph $G=(V,E)$, V is a finite set, $E \subseteq \{(u,v) \mid u,v \in V, u \neq v\} = V \times V - \{(w,w) \mid w \in V\}$
  → short notation uv for (u,v)
- $uv \neq vu$ (because of direction and strict partial order)
- $G=(V,E)$ undirected, if: $\forall u \neq v: uv \in E \Leftrightarrow vu \in E$

Def: an orientation of $G=(V,E)$ is $F \subseteq E$ such that (s.t.) $\forall uv \in E: uv \in F \Leftrightarrow vu \notin F$

Def: for a subset $F \subseteq E$, we define:
- $F^{-1} = \{vu \mid uv \in F\}$ the reversal of F
- $\hat{F} = \{uv \in E \mid uv \in F \text{ or } vu \in F\}$ the (symmetric) closure of F.

Def.: undirected graph $G=(V,E)$ is a comparability graph if it admits a transitive orientation F. We call G transitively orientable.
examples: complete graphs, path graps are comparability graphs

Obs: F is transitive orientation $\Leftrightarrow$ F^{-1} is transitive orientation

Theorem:

G is a comparability graph (comp. graph) $\Rightarrow$ G is perfect

Proof via SPGT

- observe that if G is a comp. graph, G_A is also a comp. graph $\forall A \subseteq V_G$.
- hence we show that C_t, $\overline{C_t}$, t odd, $t \geq 5$ are NOT transitively orientable.

$\Rightarrow$ odd cycle C_t, $t \geq 5$:
- take orientation F (assume trans.)
- wlog $v_1 v_2 \in F$
- F trans $\Rightarrow v_2 v_3 \notin F \Leftrightarrow v_3 v_2 \in F$

  $\Rightarrow v_4 v_3 \notin F \Leftrightarrow v_3 v_4 \in F$
- general v_i with
  - i odd must be source
  - i even must be sink
- then $v_t v_1 \in F$, $v_1 v_2 \in F$ but $v_t v_2 \notin F$, so F is NOT transitive

$\Rightarrow$ $\overline{C_t}$, $t \geq 5$ odd:
- wlog $v_1 v_3 \in F$
- since $v_4 v_3 \notin E$, we have $v_1 v_4 \in F$ (or F not trans.)

  $\Rightarrow v_1$ must be a source

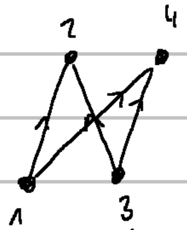
  $\Rightarrow$ all $v_3, \dots, v_{t-1}$ must be sinks $\rightarrow$ cannot have edges going out of it.

for $t \geq 6$: $\rightsquigarrow v_3 v_{t-1} \in F$ or $v_{t-1} v_3 \in F$ $\rightsquigarrow$ cannot be a pure sink anymore

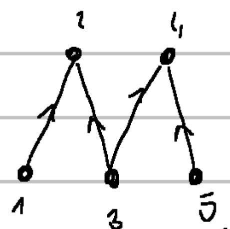
$(\overline{C_5} = C_5)$ $\rightsquigarrow$ F not orientation $\square$.

remarks: C_t, t even or $t = 3$ is comparability graph

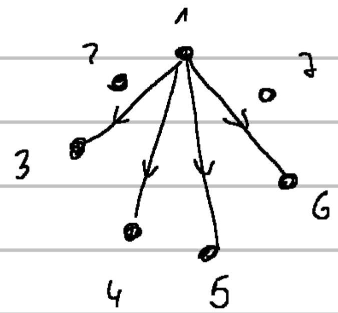
or more general: G bipartite $\Rightarrow$ G comparability graph.



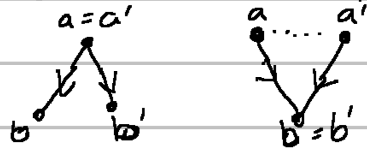
biparit
Kreis gerade



den Kreis kann man nicht schließen und zeitgleich bipartilität behalten, weil ungerade ≥ 5.



If $F \subseteq E$ transitive orientation, $ab \in F$ and $a'b' \in E$ where
$(a=a'$ and $bb' \notin E)$ or $(b=b'$ and $aa' \notin E)$ then $ab' \in F$ or $a'b \in F$



Def.: Gamma-Relation

For $ab \in E$ and $a'b' \in E$ define

$$\underset{\text{Gamma}}{ab\, \Gamma\, a'b'} \text{ if } \begin{cases} a=a' \text{ and } bb' \notin E \\ b=b' \text{ and } aa' \notin E \end{cases}$$

Obs.: $\left.\begin{array}{l} F \text{ transitive} \\ ab \in F \\ ab\,\Gamma\, a'b' \end{array}\right\} a'b' \in F$ „ab enforces/implies a'b'"

Def.: a $\overbrace{\Gamma\text{-chain}}^{\text{Gamma-chain}}$ is $a_1b_1, a_2b_2, \dots, a_kb_k$ sequence of edges (on non-necessarily distinct vertices) such that $a_ib_i\, \Gamma\, a_{i+1}b_{i+1}$ for $i=1,\dots,k-1$
we say $a_1b_1\, \Gamma^*\, a_kb_k$ (Γ^* is transitive closure of Γ)

Γ^* is an equivalence relation on E

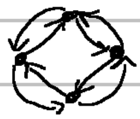
$ab\, \Gamma^*\, a'b' \Rightarrow a'b'\, \Gamma^*\, ab$

$a_1b_1\, \Gamma^*\, a_2b_2,\ a_2b_2\, \Gamma^*\, a_3b_3 \Rightarrow a_1b_1\, \Gamma^*\, a_3b_3$

Γ^* splits E into equivalence classes $\mathcal{I}(G)$ → called implication classes

Example:


C_3


C_4

C_5

C_3	C_4	C_5
$\lvert\hat{\mathcal{I}}(G)\rvert = 3$	$\lvert\hat{\mathcal{I}}(G)\rvert = 1$	$\lvert\hat{\mathcal{I}}(G)\rvert = 1$
$\lvert\mathcal{I}(G)\rvert = 6$	$\lvert\mathcal{I}(G)\rvert = 2$	$\lvert\mathcal{I}(G)\rvert = 1$

Obs.: - G comparability graph $\Rightarrow$ $\mathcal{I}(G)$ is even
- $\exists\, A \in \mathcal{I}(G),\ ab, ba \in A \Leftrightarrow G$ not a comparability graph

Def.: $A \in \mathcal{I}(G)$ then $\hat{A}$ is called a color class of G.

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

Obs.: $A \in \mathcal{I}(G) \Rightarrow A^{-1} \in \mathcal{I}(G)$

since $ab \, \Gamma^* a'b' \Rightarrow ba \, \Gamma^* b'a'$

Theorem 4.1

$A \in \mathcal{I}(G)$, F transitive orientation of $G \Rightarrow F \cap \hat{A} = A$ or $F \cap \hat{A} = A^{-1}$

(Later Theorem 4.7. is main Theorem, we need e.g. the triangle Lemma for that)

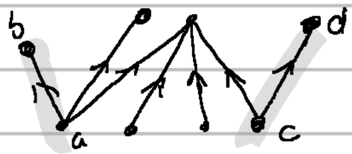
⇒ if $\mathcal{F}$ transitive orientation: $A \cap A^{-1} = \emptyset$ 30.06.25

Theorem 4.1

$\underbrace{A \in \mathcal{I}(G)}$, $\mathcal{F}$ transitive orientation of G $\Rightarrow$ $\mathcal{F} \cap \hat{A} = A$ or $\mathcal{F} \cap \hat{A} = A^{-1}$

A is an implication class, $\hat{A}$ is a color class

Obs.: $ab\,\Gamma^* cd \iff cd\,\Gamma^* ab$
$\iff ba\,\Gamma^* dc$



$A \in \mathcal{I}(G) \iff A^{-1} \in \mathcal{I}(G)$

Proof for Theorem 4.1

- let $ab \in \hat{A}$, wlog $ab \in A$ → edge in color class, that comes from A
  - Case 1: $ab \in \mathcal{F}$
    - → then $ab \in \mathcal{F} \cap \hat{A}$
    - take $cd \in A$
    - → $ab\,\Gamma^* cd$
    - along Γ-relation: $\left.\begin{matrix} a_1b_1\,\Gamma\, a_2b_2 \\ a_1b_1 \in \mathcal{F} \end{matrix}\right\} \Longrightarrow a_2b_2 \in \mathcal{F}$
    - $\Rightarrow a \in \mathcal{F}$
    - hence $A \subseteq \mathcal{F}$, $\mathcal{F} \cap \mathcal{F}^{-1} = \emptyset$ ($\mathcal{F}$ orientation)
    - $\rightsquigarrow A^{-1} \cap \mathcal{F} = \emptyset \rightsquigarrow \mathcal{F} \cap \hat{A} = A$.
  - Case 2: $ba \in \mathcal{F}$ ($ba \in A^{-1}$)
    - similarly $A^{-1} \subseteq \mathcal{F}$
    - $\rightsquigarrow A \cap \mathcal{F} = \emptyset \rightsquigarrow \mathcal{F} \cap \hat{A} = A^{-1}$ $\square$

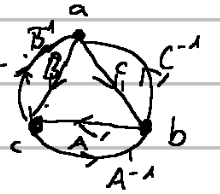
Corollary: G comp. graph, $A \in \mathcal{I}(G) \Rightarrow A \cap A^{-1} = \emptyset$ (and not $A = A^{-1}$)

Proof: G comp. graph → $\exists\, \mathcal{F}$ transitive orientation
Thm. 4.1: $\mathcal{F} \cap \hat{A} = A$ or $\mathcal{F} \cap \hat{A} = A^{-1}$ but $\mathcal{F} \cap \mathcal{F}^{-1} = \emptyset$
hence $A \cap A^{-1} = \emptyset$. $\square$

If all implication classes are nice, G is comp. graph, if there are implication classes, that are not nice, G not comp. graph.

Triangle Lemma → also look at slides for visual understanding.

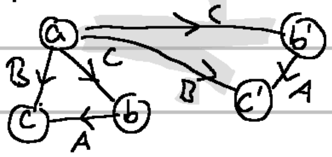
- G graph, $A, B, C \in \mathcal{I}(G)$
- triangle abc in G



class A contains the edge, that does NOT contain vertex a. Same for classes B and C.

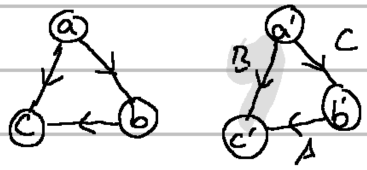
- assumption: $A \neq B$, $A \neq C^{-1}$.

(i)



$\left.\begin{array}{l} b'c' \in A \end{array}\right\} \Rightarrow \begin{array}{l} ab' \in C \\ ac' \in B \end{array}$

(ii)

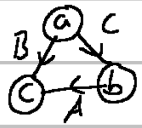


$\left.\begin{array}{l} a'b' \in C \\ b'c' \in A \end{array}\right\} \Rightarrow a'c' \in B$

important: $B^{-1} = C$, $A = C$, … is possible, we only assume that $A \neq B$, $A \neq C^{-1}$
$a' = b$, $b' = c$, … is also possible

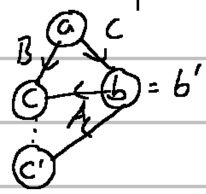
Proof:

(i):



- it is enough to consider in Γ-chain $bc \,\Gamma\, b'c'$
- two cases: 1: $b = b'$, $cc' \notin E$
  2: $c = c'$, $bb' \notin E$

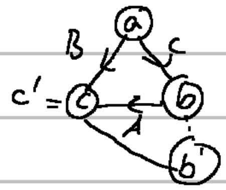
case 1: $b = b'$, $cc' \notin E$



— edge
… non-edge

- Obs: $c' \neq A$ since $cc' \notin E$
- if $ac \notin E$ then $\underbrace{ba}_{\in C^{-1}} \,\Gamma\, \underbrace{bc'}_{\in A}$ thus $C^{-1} = A$ } contradiction ↯ to $A \neq C^{-1}$
- hence $ac \in E$ and $ac \,\Gamma\, ac'$, thus $ac' \in B$

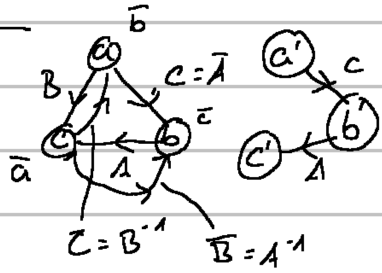
case 2: $c = c'$, $bb' \notin E$



} similar to case 1

- if $ab' \notin E$ then $\underbrace{b'c}_{\in A} \,\Gamma\, \underbrace{ac}_{\in B}$ ⇒ contradiction to $A \neq B$
- so $ab' \in E$ and $ab \,\Gamma\, ab'$, thus $ab' \in C$

Proof

(ii) 

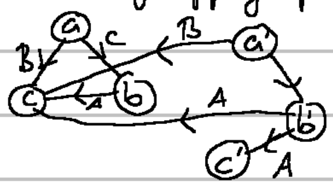
apply (i) to $\bar a, \bar b, \bar c$, $\bar A, \bar B, \bar C$

we need: $C = \bar A \neq \bar B = A^{-1}$, so $C \neq A^{-1}$ } we know that

we need: { $C = \bar A \neq \bar C^{-1} = (\bar B^{-1})^{-1} = B$, so $C \neq B$

↳ we don't know that.

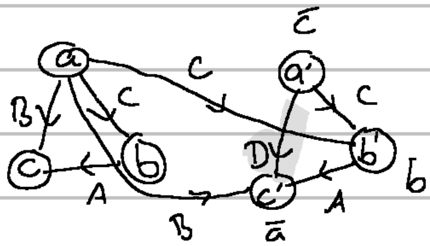
case $B \neq C$: ⇝ may apply part (i)



$\Rightarrow$ $a'c \in B$, $b'c \in A$

- again part (i) to triangle $a'b'c$ gives $a'c' \in B$

case $B = C$:



$A \neq B$, $A \neq C^{-1}$

part (i): gives $ab' \in C$, $ac' \in B$

if $a'c' \notin E$, then $\underbrace{b'a'}_{\in C^{-1}} \; \Gamma' \; \underbrace{b'c'}_{\in A}$ } ⊗

⊗ Contradiction to $A \neq C^{-1}$.

so $a'c' \in E$.

- Let $a'c' \in D \in \mathcal{I}(G)$, $D \neq B$ (otherwise we are done)

- part (i) to $\bar a \bar b \bar c$, $\bar A = C^{-1}$, $\bar B = D^{-1}$, $\bar C = A^{-1}$

Verify that • $\bar A \neq \bar B$: $B^{-1} = C^{-1} = \bar A \neq \bar B = D^{-1}$

• $\bar A \neq \bar C^{-1}$: $C^{-1} = \bar A \neq \bar C^{-1} = A$ ↝ we know that

add $ba \in C^{-1} = \bar A \neq \bar C^{-1} = A$ ✓ ⇝ gives $ac' \in D$, but then $B = D$ □.

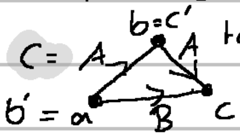
Theorem 4.4

If $A \in \mathcal{I}(G) \rightarrow A$ is implication class, there are two cases:

1) $A = A^{-1}$

2) $A \cap A^{-1} = \emptyset$ and A, A^{-1} transitive

(we knew $A = A^{-1}$ or $A \cap A^{-1} = \emptyset$, in case $A \cap A^{-1}$ we show, that A is transitive)



to show $\Longrightarrow ac \in A$

$\quad\quad\quad\quad A \quad\quad A^{-1}$

- if $ac \notin E$, then $ab \,\Gamma\, cb \Rightarrow$ then $A = A^{-1}$ ↯ contradicts case $A \cap A^{-1} = \emptyset$

$\rightsquigarrow ac \in E$ so $ac \in B$, $B \in \mathcal{I}(G)$ to show $A = B$.

- Assume $B \neq A$: - call Triangle Lemma

  Note that: $A \neq B$ and $A \neq C^{-1} = A^{-1}$ (because $A \cap A^{-1} = \emptyset$)

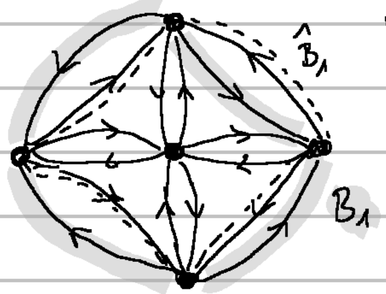
  part (i) with $b'c' = ab$ (new base)

  $\rightsquigarrow$ we get: $\left.\begin{array}{l} ac' \in B \\ ab \in A \end{array}\right\} \Rightarrow A = B \rightarrow$ contradicts assumption $A \neq B$ ↯ $\square$

  for sake of contradiction

(because $ab \,\Gamma^*\, bc$, also $ab \,\Gamma^*\, ac$)

$\Rightarrow$ A transitive and therefore A^{-1} also transitive.



take $B_1 \in \mathcal{I}(G)$

remove $\hat{B}_1$

take $B_2 \in \mathcal{I}(G - B_1)$

$\Rightarrow$ recognition of comparability graphs, algorithm on slides

algorithm: F empty, $i = 1$, Take an subset of E. while subset not empty, take random edge, find implication class of edge, test if $B \cap B^{-1} = \emptyset$

if not $\Rightarrow$ NOT a comparability graph

if $B \cap B^{-1} = \emptyset$, then continue with search

if G is comparability graph, you get sequence $B_1, B_2, \dots, B_4$

Def: $[B_1, B_2, \dots, B_k]$ is a G-decomposition if:

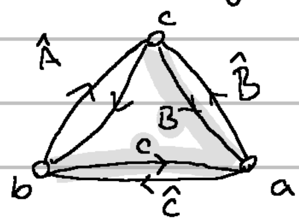
1) $B_1 + B_2 + \dots + B_k = E$ and
2) $B_i \in \mathcal{I}(\hat{B}_i + \dots + \hat{B}_k)$ for $i = 1, \dots, k$
   ↳ from $\hat{B}_i$ up to $\hat{B}_k$

Note: Algorithm 7 computes a G-decomposition or stops with result „NOT a comparability graph".
⇒ if G is a comparability graph, if Algo 7 computes a G-decomposition
(to show that, we need to understand, what happens to implication classes of original graph when we remove color classes) [NOTE: ALL graphs have a G-decomposition, but for comparability graphs, all G-decompositions are „nice"]

Theorem 4.6:

$A \in \mathcal{I}(G)$, $D \in \mathcal{I}(E - \hat{A})$ } ⇒ (A implication class of G, D implication class of Edges with color class $\hat{A}$ removed).
then:

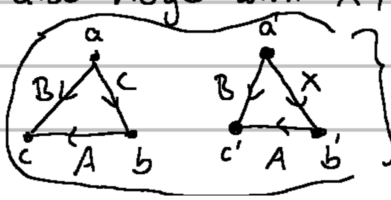
(i) $D \in \mathcal{I}(G)$ and $A \in \mathcal{I}(E - \hat{D})$
(removing color class of A keeps D intact and removing color class of D keeps A intact)
or (ii) $D = B + C$, $B, C \in \mathcal{I}(G)$ and ∃ rainbow-triangle $\hat{A}, \hat{B}, \hat{C}$
(rainbow-triangle: if I remove A then B and C merge)



$A \in \mathcal{I}(G)$, $D \in \mathcal{I}(E - \hat{A})$
rainbow-triangle

Proof:
- removing $\hat{A}$ introduces new non-edges (removes existing edges)
  ⇝ introduces new Γ-relations ⇒ implication classes can merge
⇒ $D \in \mathcal{I}(G - \hat{A})$ then D is a disjoint union of some previous existing implication classes

Case 1: $D = B + C + \dots$, $B, C \in \mathcal{I}(G)$ } goal: show, that max. 2 can form D
⇝ ∃ rainbow-triangle $\hat{A}, \hat{B}, \hat{C}$ → B and C merge.
- if B would also merge with X, there would also be a rainbow-triangle $\hat{A}, \hat{B}, \hat{X}$



triangle Lemma part (ii): $\hat{X} = \hat{C}$, so it is NOT different to C, only B and C merge ⇒ $D = B + C$

↳ second szenario

case 2: $\mathcal{D} \in \mathcal{I}(G)$.

(first scenario) - by case 1: every implication class of $\mathcal{I}(G-\hat{\mathcal{D}})$ is union of at most 2 implication classes of $\mathcal{I}(G)$.

- if A merges with X in $G-\hat{\mathcal{D}}$, then there is a rainbow-triangle of $\hat{A}, \hat{\mathcal{D}}, \hat{X}$
- then $\mathcal{D}$ merges with X or X^{-1} in $G-\hat{A}$ ↯
  $\Rightarrow A \in \mathcal{I}(G-\hat{\mathcal{D}})$

Theorem 4.7:

(i) G comparability graph
$\Leftrightarrow$ (ii) $A \cap A^{-1} = \emptyset \quad \forall A \in \mathcal{I}(G)$
$\Leftrightarrow$ (iii) $\forall$ G-decompositions $[B_1, \dots, B_k]$ has $B_i \cap B_i^{-1} = \emptyset \quad \forall i \in [k]$

Proof:

(i) $\Rightarrow$ (ii): done (Theorem 4.1)

(ii) $\Rightarrow$ (iii): $[B_1, \dots, B_k]$ any G-decomposition.

induction on k:

- $k=1$: $B_1 \in \mathcal{I}(G)$, so $B_1 \cap B_1^{-1} = \emptyset$ by (ii)
- $k \geq 2$: - Again $B_1 \cap B_1^{-1} = \emptyset$ by (ii)
  - $[B_2, \dots, B_k]$ is a G-decomposition of $G-\hat{B}_1$

B_1 is implication class in G and (ii) says, that it is nice. Removing it does not change anything, so All B_i in G-decomposition (which is only B_1) are nice.

- verify: $\mathcal{D} \cap \mathcal{D}^{-1} = \emptyset \quad \forall \mathcal{D} \in \mathcal{I}(G-\hat{B}_1)$
- Thm. 4.6: $\mathcal{D} \in \mathcal{I}(G)$ and $\mathcal{D} \cap \mathcal{D}^{-1} = \emptyset$ by (ii)
  or $\mathcal{D} = B + C$ for $B, C \in \mathcal{I}(G)$
  $\rightarrow \mathcal{D} \cap \mathcal{D}^{-1} = (B+C) \cap (B+C)^{-1} = (B+C) \cap (B^{-1} + C^{-1}) =$

$$= \underbrace{(B \cap B^{-1})}_{= \emptyset \text{ by (ii)}} \cup \underbrace{(B \cap C^{-1})} \cup \underbrace{(C \cap B^{-1})} \cup \underbrace{(C \cap C^{-1})}_{= \emptyset \text{ by (ii)}} =$$

$B \neq C^{-1}$, $C \neq B^{-1}$
since rainbow-triangle $\hat{B}_1, \hat{B}, \hat{C}$

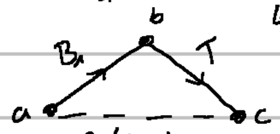
$= \emptyset$

- $\rightsquigarrow$ (ii) holds for $G-\hat{B}_1$. by induction $B_i \cap B_i^{-1} = \emptyset \quad \forall i \geq 2$.

(iii) $\Rightarrow$ (i): look next page

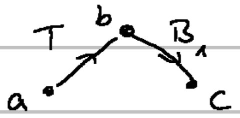
<u>(iii) ⇒ (i):</u> induction on k: $[B_1, \dots, B_k]$

- $k = 1$: $B_1 \cap B_1^{-1} = \emptyset$ $\rightsquigarrow$ by Thm. 4.4: B_1 is transitive and an orientation
  $\Rightarrow G$ is comparability graph
- $k \geq 2$: - $[B_2, \dots, B_k]$ is G-decomposition of $G - \hat{B}_1$, fulfills (iii)
  $\rightsquigarrow$ by induction: $G - \hat{B}_1$ has transitive orientation T.
  - Claim: $B_1 + T$ is transitive orientation of G
    ↳ not clear ↳ clear



↳ we want edge here, that is oriented from a to c

- if this would be a non-Edge, then $ab \overset{B_1}{\Gamma} cb \Rightarrow cb \in B_1$ ↯ $bc \in T$
  so $ac \in E$
- right direction: • $ca \in T \Rightarrow T$ not transitive ↯
  • $ca \in B_1 \Rightarrow B_1$ not transitive ↯
  $\Bigg\}$ $ac \in B_1$ or $ac \in T$ $\Rightarrow B_1 + T$ transitive



$\Big\}$ analog $\Rightarrow T + B_1$ transitive

$\square$

<u>Corolary:</u> Algorithm 7 determines correctly whether G is a comparability graph in $O(\Delta(G) \cdot |E| + |V|)$ $\rightarrow$ polynomial

Theorem: 14.07.25

Algo 8 computes $\chi(G), \omega(G)$ correctly for G comparability graph

Proof:

- <u>h is proper coloring</u>, i.e. $\forall uv \in E(G) \to$ show that $h(u) \neq h(v)$:
  - wlog $uv \in F \rightsquigarrow \sigma(u) < \sigma(v)$

    $\rightsquigarrow h(u)$ is set already when $v = \sigma(i)$

    - $h(v) = 1 + \max\{h(w) \mid wv \in F\} \geq 1 + h(u)$ ✓

      $\Rightarrow \max\limits_{v \in V} h(v) \geq \chi(h)$

- <u>C is a clique:</u>
  - let $C = \{w_\chi, w_{\chi-1}, \dots, w_1\} \rightsquigarrow \chi: h(w_\chi) = 1 + \max\{h(v) \mid v w_\chi \in F\}$

    $\rightsquigarrow h(w_{\chi-1}) = \chi - 1, \ w_{\chi-1} w_\chi \in F$

    $\rightsquigarrow h(v_{\chi-i}) = \chi - i, \ w_{\chi-i-1} w_{\chi-i} \in F$

    C is a directional path in F

  - by transitivity of F, C is a clique ✓

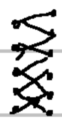
$$\Rightarrow \begin{array}{lcl} \chi = |C| & \leq & \omega(G) \\ \| & & \rotatebox[origin=c]{90}{$\leq$} \\ \max\limits_{v \in V} h(v) & \geq & \chi(G) \end{array} \qquad \square$$

runtime of Algo 8: linear $O(|V| + |E|)$, exept line 1

Theorem: (König)

Algo 9 computes $\alpha(G), k(G)$ for G comp. graph in polynomial time

special case: G bipartite



Def.: $M \subseteq E_G$ is a Matching if $\forall v \in V \ \exists^{\leq 1} e \in M: v \in e$

$k(G) \leq |V| - \max\{|M| : M \text{ matching}\}$ (Case bipartite: equality)

Def.: $S \subseteq V_G$ is a vertex cover if $\forall e \in E \ \exists^{\geq 1} v \in S: v \in e$. holds for

$k(G) \geq \alpha(G) = |V| - \min\{|S| : S \text{ vertex cover}\}$ - - - - - - - - every graph

Note: 1) k is the number of cliques in G

2) S vertex cover $\Leftrightarrow V - S$ independent

slides

v, w consecutive in clique cover $\overset{\text{Rule}}{\Longleftrightarrow}$ $v'w'' \in M$

clique cover $V_1 + \dots + V_k$ of G

$\Leftrightarrow$ k cliques in G (partitioning V_G)

$\Leftrightarrow$ $2k$ starts and ends of paths

$\Leftrightarrow$ $2k$ vertices of B are unmatched

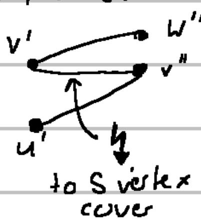
$\Leftrightarrow$ $|V_G| - k = |M|$

$\leadsto k(G) = |V| - \max\{|M| : M \text{ matching in } B\} \overset{\text{König}}{=} |V| - \min\{|S| : S \text{ vertex cover in } B\}$

We get vertex cover S of B
- $\forall v \in S$: $S - v$ is NOT a vertex cover

<u>Obs.:</u> $|S \cap \{v', v''\}| \leq 1 \quad \forall v \in V_G$

Assume not:



- Since $S - v'$ not vertex cover $\leadsto \exists w \in V_G$, $w \notin S$, $vw \in F$
- Since $S - v''$ not a vertex cover $\leadsto \exists u \in V_G$, $u' \notin S$, $uv \in F$

$\Rightarrow uw \in F \Rightarrow u'w'' \in E_B$

Hence $Y = \{v \in V_G : S \cap \{v', v''\} = \emptyset\}$ has exactly $|V| - |S| = |V| - |M| = k(G)$ elements

<u>Obs.:</u> Y is independent set in G.

- $vw \in E_G \leadsto$ wlog $vu \in F$

$\Rightarrow v'w'' \in E_B$ but $S \cap \{v', w''\} = \emptyset$ ↯ to S vertex cover

Hence $\alpha(G) \geq |Y| = k(G) \overset{\forall G}{\geq} \alpha(G) \Rightarrow |Y| = \alpha(G)$

<u>Algo 9</u>

	runtime
- compute transitive orientation F	$O((\lvert V\rvert + \lvert E\rvert)^2)$
- compute bipartite graph B	$O(\lvert V\rvert + \lvert E\rvert)$
- compute max. matching M	$O(\lvert E\rvert^{1,5})$
- compute min. vertex cover S	$O(\lvert V\rvert + \lvert E\rvert)$
- derive clique cover on $\lvert V\rvert - \lvert M\rvert$	
- derive i-set on $\lvert V\rvert - \lvert S\rvert$	

□


induced C_4
P_4 not splitgraph

22.07.25

Theorem 5.3

(i) G is chordal and $\overline{G}$ is chordal (split graph)

$\Leftrightarrow$ (ii) $V = K + S$ with K clique and S independent set

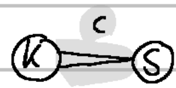
$\Leftrightarrow$ (iii) $C_4, C_5 \notin_{ind} G$ and $C_4 \notin_{ind} \overline{G}$

(C_4 and C_5 not induced subgraphs of G and C_4 not indused subgraph of $\overline{G}$)

Proof of Theorem 5.3.:

(ii) $\Rightarrow$ (i): - $V = K + S$; K clique, S independent set

- C cycle of length ≥ 4 in G (↯ G chordal)



$\rightarrow$ if $V(C) \cap S = \emptyset$ then C has a chord since K is a clique

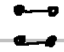
$\rightarrow$ if $C = [v_1, v_2, v_3, v_4, \ldots]$, $v_2 \in S$ then $v_1, v_3 \in K$ since S is an independent set

$\Rightarrow v_1 v_3 \in E$ i.e. a chord since K is a clique

$\rightsquigarrow$ G is chordal $\rightsquigarrow$ $\overline{G}$ is chordal by the same reason (for $\overline{G}$: K is independent set, S is a clique)

(i) $\Rightarrow$ (iii): clear (have we shown before)

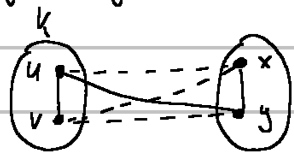
(iii) $\Rightarrow$ (ii): we find split K, S

- choose K a maximum clique with $S = V - K$, G_S has fewest edges
- Assume G_S has an edge xy, we shall find C_4 or $C_5 \subseteq_{ind} G$ or $C_4 \subseteq_{ind} \overline{G}$ (so $2K_2 \subseteq_{ind} G$) $2K_2$:  $\overline{2K_2}$: 
- K maximum $\rightsquigarrow \exists u \in K : ux \notin E$

$\rightsquigarrow \exists v \in K : vy \notin E$

$\rightarrow$ if $u = v$ for all choices, then $K - u + \{x, y\}$ larger clique ↯ K maximum clique

- so $u \neq v$



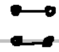
$\rightarrow$ if $vx, uy \in E$ then $C_4 \subseteq_{ind} G$ ✓

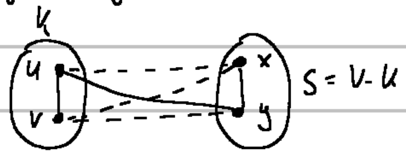
$\rightarrow$ if $vx, uy \notin E$ then $2K_2 \subseteq_{ind} G$ ✓

$\Rightarrow$ so wlog: $vx \notin E$, $uy \in E$

[continued next page]

(iii) $\Rightarrow$ (ii): we find split K, S

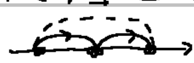
- choose K a maximum clique with $S = V - K$, G_S has fewest edges
- Assume G_S has an edge xy, we shall find C_4 or $C_5 \subseteq_{ind} G$ or $C_4 \subseteq_{ind} \overline{G}$ (so $2K_2 \subseteq_{ind} G$) $2K_2$:  $\overline{2K_2}$: 
- K maximum $\rightsquigarrow \exists u \in K : ux \notin E$
  $\rightsquigarrow \exists v \in K : vy \notin E$
- $\rightarrow$ if $u = v$ for all choices, then $K - u + \{x, y\}$ larger clique $\lightning$ K maximum clique
- so $u \neq v$



- $\rightarrow$ if $vx, uy \in E$ then $C_4 \subseteq_{ind} G$ ✓
- $\rightarrow$ if $vx, uy \notin E$ then $2K_2 \subseteq_{ind} G$ ✓
- $\Rightarrow$ so wlog: $vx \notin E$, $uy \in E$
- consider $K' = K - v + y$ (want to show, that K' is a clique)
- $w \in K - \{u, v\}$, $wy \notin E$ $\rightsquigarrow$ $wx \notin E \Rightarrow G_{\{v,w,x,y\}} = 2K_2$ ✓
  $\rightsquigarrow$ $wx \in E \Rightarrow G_{\{v,w,x,y\}} = C_4$ ✓
- so $\forall w \in K - \{u, v\}$: $wy \in E$, i.e. K' is a clique, $|K'| = |K|$
- want to show that $G_{V-K'}$ has fewer edges than G_{V-K}
- to show $x \in |Adj(y) \cap S| > x \notin |Adj(v) \cap S|$
- assume $t \in S$, $tv \in E$, $ty \notin E$ $\rightsquigarrow$ if $tx \notin E$ then $G_{\{x,y,v,t\}} = 2K_2 \Rightarrow$ so $tx \in E$
- $\rightarrow$ if $tu \notin E$ then $G_{\{u,v,x,y,t\}} = C_5$ ✓ $\Rightarrow$ so no such $t \in S$ exists.
- $\rightarrow$ if $tu \in E$ then $G_{\{u,t,y,x\}} = C_4$ ✓
- $\rightsquigarrow |Adj(y) \cap S| > |Adj(v) \cap S| \rightsquigarrow G_{V-K'}$ has fewer edges then G_{V-K} $\lightning$ to choice of K $\square$.

G chordal $\Leftrightarrow \exists\, \sigma$ w/o (without)

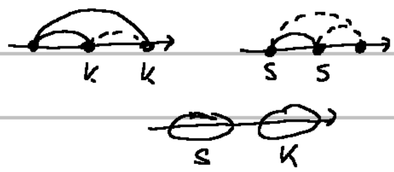
G comparability graph $\Leftrightarrow \exists\, \sigma$ w/o



G co-chordal $\Leftrightarrow \exists\, \sigma'$ w/o

$\overline{G}$ comparability graph $\Leftrightarrow \exists\, \sigma'$ w/o

G split graph $\Leftrightarrow \exists\, \sigma$ w/o



Theorem:

For every (undirected) graph $G = (V, E)$

(i) G and $\bar{G}$ are comparability graphs

$\Leftrightarrow$ (ii) There exists a vertex ordering $\mathcal{O}$ of G w/o and w/o

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

$(uv \in E)$ iff $(u_x < v_x \Leftrightarrow u_y < v_y)$

Proof:

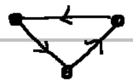
<u>(i) ⇒ (ii):</u> - G has transitive orientation (V, F_1)

- $\bar{G}$ has transitive orientation (V, F_2)

$\}$ $F = F_1 + F_2$ is an orientation of complete graph on V.

Claim: F_1, F_2 transitive $\Rightarrow F = F_1 + F_2$ is transitive

- F orientation of complete graph is transitive $\Leftrightarrow F$ is acyclic



not transitive
→ directed cycle

longer directed cycles contain directed triangles $\rightsquigarrow$ not transitive

- F has directed triangles (F_i not transitive) or (F_i not transitive) * $F_{(i+1) \bmod 2}$
- Let $\mathcal{O}$ be top. ordering of $F = F_1 + F_2$

pattern ↯ F_1 trans. and ↯ F_2 trans.

<u>(ii) ⇒ (iii):</u> - given $\mathcal{O}$ w/o ,

- obtain transitive orientation F_1 of G, F_2 of $\bar{G}$ by orienting left-to-right
- take $\mathcal{O}_x = \mathcal{O}$ as order of x-coordinates of points for each vertex in V.
- $F_1 + F_2^{-1}$ is also transitive orientation of complete graph

$\rightsquigarrow$ 2nd ordering $\mathcal{O}_y$ as top. ordering of $F_1 + F_2^{-1}$

$uv \in E$, wlog $uv \in F_1 \Rightarrow u_x < v_x$ and $u_y < v_y$

$uv \notin E$, wlog $uv \in F_2 \Rightarrow u_x < v_x$ but $vu \in F_2^{-1} \Rightarrow u_y > v_y$

Proof: continued 

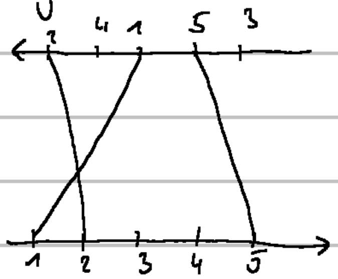
(iii) $\Rightarrow$ (i): - given embedding of V in the plane

- orientate $uv \in E$ from u to $v \Leftrightarrow u$ is bottom-~~left~~ right of v

↳ this is transitive $\Rightarrow G$ is comparability graph
$\Rightarrow \overline{G}$ — " —

- given ordering π of the first $[n]$ numbers
- $G = G_\pi$ $\quad V(G) = [n]$
- $ij \in E \Leftrightarrow (i-j)(\pi_i - \pi_j) < 0$

Recognition in Linear Time $\mathcal{X}, \omega$ in $O(|V|+|E|)$



segments, intersection

- Given intervals $I_1, \dots, I_n$; $I_i = (x_i, y_i)$ sorted, so $x_1 \le x_2 \le \dots \le x_n$

- Goal: find the number of minimal translations, such that they do not intersect
we want (i) $x_1' \le \dots \le x_n'$
and (ii) $y_i' < x_{i+1}'$ $\forall i \in [n-1]$

• construct the conflict graph G with $V(G) = \{I_1, \dots, I_n\}$
$I_i I_j \Leftrightarrow x_j - y_i < \sum_{i<k<j} (y_k - x_k)$ } sum of the lengths of the intervals between i and j

Show that G is permutation graph: 

↳ show that we have those two patterns

$\Rightarrow$ shows, that the forbidden patterns  and  are not there

solution is minimal set of intervals that are moved
$\Leftrightarrow$ maximum set of intervals that are not moved
$\Leftrightarrow$ maximum independent set in G

Theorem 7.1 (on slides)

For every $G=(V,E)$ the following are equivalent

(i) G is an interval graph

(ii) $\exists$ vertex ordering σ without

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

(iv) $C_4 \notin_{ind} G$ and $\bar{G}$ is a comparability graph

(v) There exists an ordering $A_1, \dots, A_x$ of the inclusion-maximal cliques in G such that $\forall v \in V$ the numbers in $\{i \mid v \in A_i\}$ are consecutive in $\{1, \dots, x\}$

Proof:

(i) => (ii): - look at interval representation

→ wlog all endpoints are distinct

- define σ as the left to right ordering of endpoints

- let $u <_\sigma v <_\sigma w$ with $uw \in E$  => $vw \in E$

(ii) => (iii): σ has no or => both forbidden by σ w/o

=> chordal graphs characterised by not having those

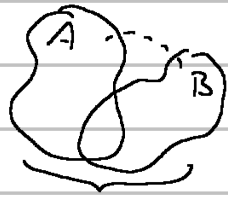
=> comparability graphs characterized by not having , what is complement of the => $\bar{G}$ comparability graph

(iii) => (iv): trivial

(iv) => (v): - C_4 is not an induced subgraph of G and $\bar{G}$ comp. graph 

- $2K_2$ not induced subgraph of $\bar{G}$

- Let F be transitive orientation of $\bar{G}$

- let A, B be maximal cliques 

- there is a non-edge ab between $A-B$ and $B-A$

- if $ab \in F$, we say $A < B$; if $ba \in F$, we say $B < A$

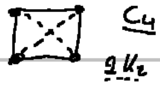
↳ cases are in old script

→ with cases in old script, we show that if $A < B$, then NOT $B < A$

↳ $<$ is well defined.

- still need to show, that $<$ is acyclic

(iv) ⇒ (v): - C_4 is not an induced subgraph of G and $\bar{G}$ comp. graph



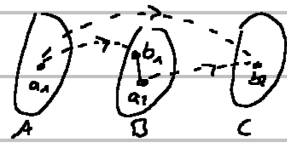
- $2K_2$ not induced subgraph of $\bar{G}$
- Let F be transitive orientation of $\bar{G}$
- let A, B be maximal cliques
- there is a non-edge ab between $A-B$ and $B-A$
- if $ab \in F$, we say $A < B$; if $ba \in F$, we say $B < A$

↳ cases are in old script

→ with cases in old script, we show that if $A < B$, then NOT $B < A$

↳ $<$ is well defined.

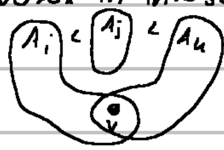
- still need to show, that $<$ is acyclic ⇒ Goal
- let $A < B < C$; goal: $A < C$ (so $<$ is acyclic)



- let $a_1 b_1 \in F$, $a_1 \in A$, $b_1 \in B$
- let $a_2 b_2 \in F$, $a_2 \in B$, $b_2 \in C$
  - $b_2 \notin A$
  - if $b_1 = a_2$ → we also have: $a_1 b_2 \in F$ ⇒ $A < C$ ✔
  - assume $b_1 \neq a_2$ → $b_1 a_2 \in E$ (because B is clique) and $a_1 a_2 \in E$, $b_1 b_2 \in E$

    ↳ C_4 not induced subgraph of G ⇒ $a_1 b_2 \notin G$

    ⇒ because F transitive, we know that $a_1 b_2 \in F$ ⇒ $A < C$
- There is a total order $A_1 < \dots < A_x$ on maximal cliques
- Let $v \in A_i \cap A_k$ with $i < j < k$ (v is vertex in intersection of A_i and A_k)
- Goal: show that $v \in A_j$
- Assume that $v \notin A_j$



↳ then there is vertex $w \in A_j$ with $vw \notin E$

- if $vw \in F$ ⇒ $A_k < A_j$ ↯ as we said $i < j < k$ before
- so $wv \in F$ ⇒ $A_j < A_i$ ↯ — " —

⇒ so Assumption wrong and $v \in A_j$

(v) ⇒ (i): - Given $A_1 < \dots < A_x$ ordering on maximum cliques

- $\{i \mid v \in A_i\}$ is an interval
- Let I_v be the smallest interval, such that $\{i \mid v \in A_i\} \subseteq I_v$
- $vw \in E \iff \exists i : vw \in A_i \iff I_v \cap I_w = \emptyset$

⇒ G is an interval graph

Cops-and-Robber-game: you can always proove, that there is either a strategy for the rober to continue infinitely or the Cops can win for k Cops

Note: Interval graphs special kind of chordal graphs; there is a connection to matrices for interval graphs

Interval graph check: G is chordal and $\overline{G}$ is a comparability graph

1991 Simon algorithm to recognize interval graphs

↳ compute PES, then compute second lexBFS using first PES to get nicer ordering, repeat with second and third ordering. On the fourth one, you can show, that a graph is an interval graph

Cops-And-Robber-game part 2: robber can now move on a path instead of one edge (but not through Cops) -> Robber is fast Fast-Robber

Cops now can now arbitrarily move -> Cops have helicopter Helicopter-Cop

While cop is in the air, robber can move

⇒ this variant: number of Cops needed to win = treewidth + 1

asteroidal triples: triples of vertices a, b, c. There is a path from a to b, that avoids the neighbourhood of c AND a path from a to c, that avoids the neighbourhood of b AND a path from b to c that avoids the neighbourhood of a.

Cops-and-Robber-game part 3: ordinary game from beginning.

For any numbers of cops there is a comp. graph, where the robber wins

For chordal graphs one cop is enough, if $\overline{G}$ is chordal (so G is co-chordal) you need two cops