Lecture notes for Algorithmic Graph Theory

Sven Geißler ✉
Karlsruhe Institute of Technology, Germany

1 Preliminaries

We begin these lecture notes by defining the basic structures used in this course.

▶ **Definition 1.** *A* graph $G = (V, E)$ *consist of a finite vertex set* V *with* $|V| \geq 1$ *and a set of edges* $E \subseteq \{\{u, v\}|u, v \in V, u \neq v\} = \binom{V}{2}$.

Note, that this definition allows for neither parallel edges nor loops and thus can be seen as a undirected simple graph. In the following we use the simplified notation $\{u, v\} = uv$ for edges. Note, that this implies $uv = vu$.

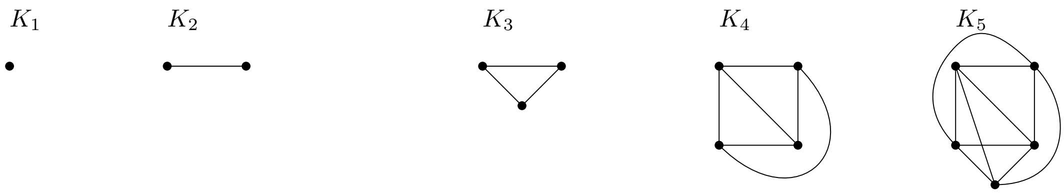
2 Introduction

In this section we introduce some simple graph families as well as the parameters studied in this course. Furthermore, the graph class of *perfect graphs* and their two most important structural results are introduced.

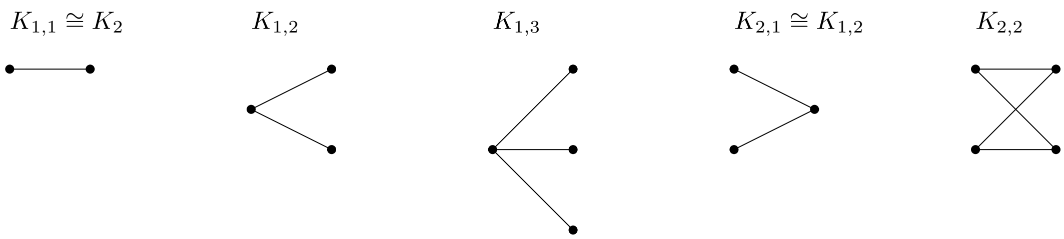
2.1 Important graphs

In this section we introduce some graph families used throughout this lecture.

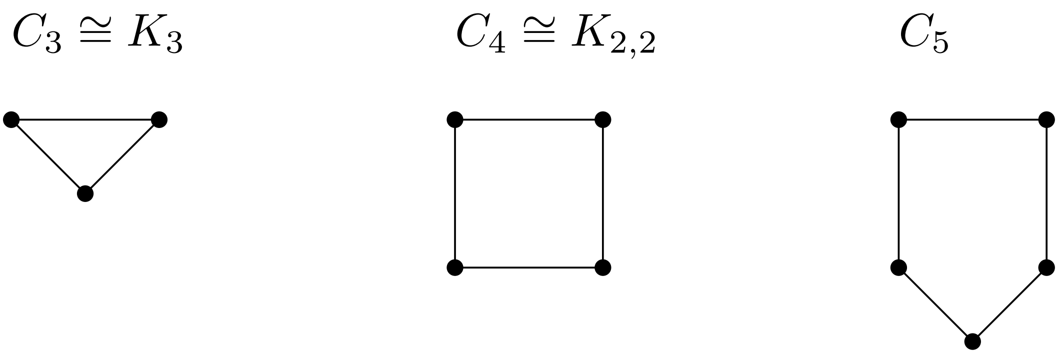
- For $n \geq 1$ we define $K_n = \left([n], \binom{[n]}{2}\right)$ as the *complete graph* on n vertices. Here, we used $[n] = \{1, \ldots, n\}$. So using the naturally defined functions V and E, we have: $V(K_n) = [n]$ and $E(K_n) = \binom{[n]}{2}$.



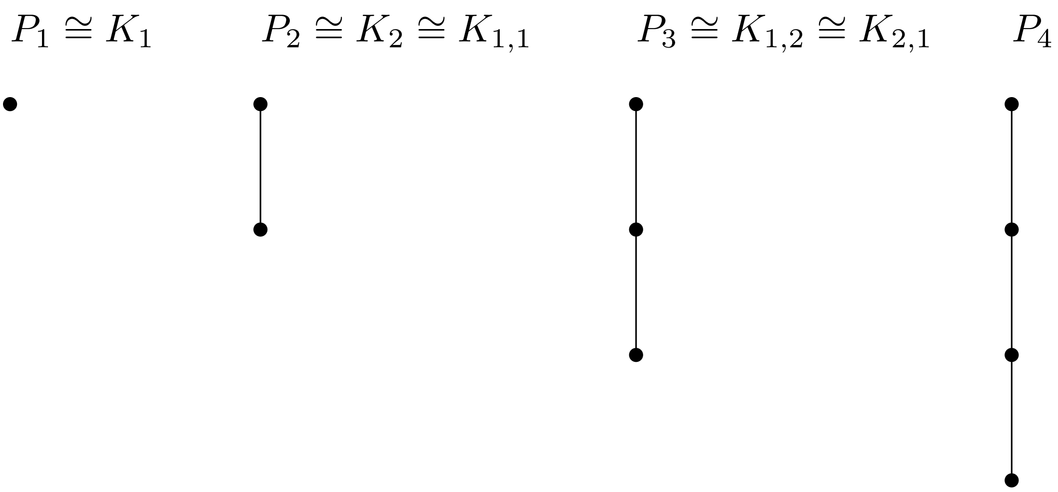
- For $n, m \geq 1$ we define $K_{n,m} = (\{a_1, \ldots, a_n\} \cup \{b_1, \ldots, b_m\}, \{a_i b_j | i \in [n], j \in [m]\})$ as the *complete bipartite graph* on $n + m$ vertices.



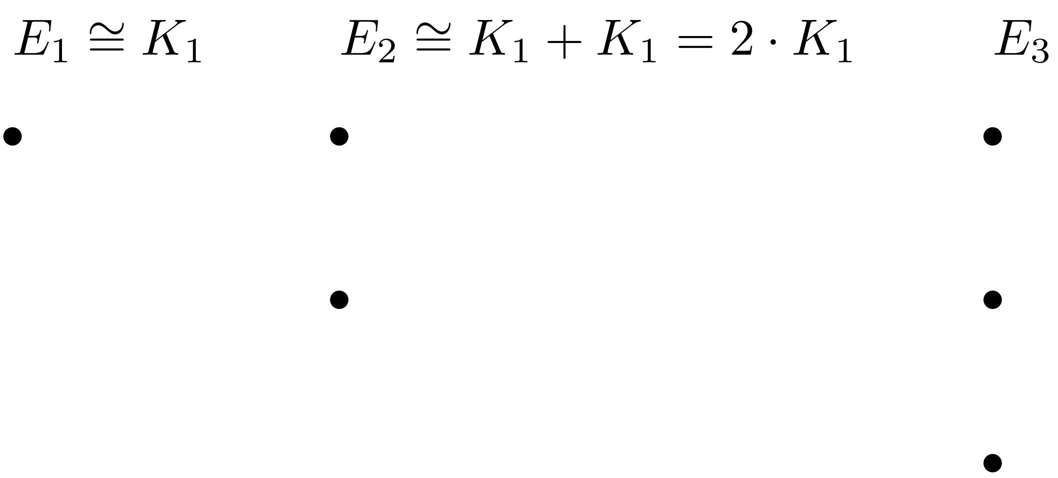
- For $n \geq 3$ we define $C_n = ([n], \{\{i, i+1\}|i \in [n-1]\} \cup \{1n\})$ as the *cycle* on n vertices.



- For $n \geq 1$ we define $P_n = ([n], \{\{i, i+1\} | i \in [n-1]\})$ as the *path* on n vertices. Note that $[0] = \emptyset$ and $P_n = C_n - 1n$ for $n \geq 3$.



- For $n \geq 1$ we define $E_n = ([n], \emptyset)$ as the *empty graph* on n vertices.

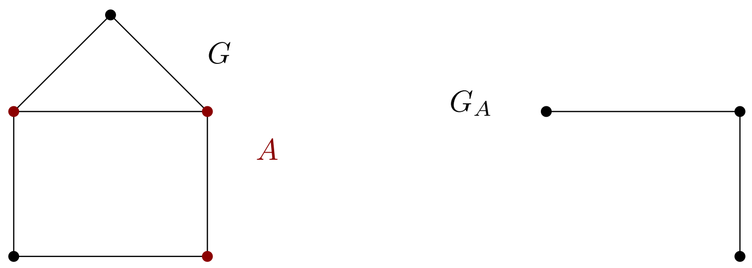


2.2 The parameters

We continue by introducing four parameters studied in this lecture. To formally define the parameters we need some terminology, which we introduce first.

▶ **Definition 2.** *For a graph $G = (V, E)$ and a vertex subset $A \subseteq V$ the* induced subgraph G_A *is defined by* $V(G_A) = A$ *and* $E(G_A) = \{uv \in E | u, v \in A\}$.

We use the notation $G_A \subseteq G$.



We denote the *disjoint union of sets* as $A + B = A \cup B$ but if $A \cap B = \emptyset$. We use this in the next definition.

▶ **Definition 3.** *A* partition *in t parts, $t \geq 1$, of a set V is $V_1 + \cdots + V_t = V$.*

We are now ready to introduce our parameters.

▶ **Definition 4.** *For a graph $G = (V, E)$, a set $A \subseteq V$ and a partition $V_1 + \cdots + V_t = V$ we define:*

- *A* clique *if G_A is a complete graph.*
- *A* independent set *if G_A is an empty graph.*
- clique number $\omega(G) = \max\{|A| : A \subseteq V(G) \textit{is clique}\}$.
- independence number $\alpha(G) = \max\{|A| : A \subseteq V(G) \textit{is independent set}\}$.
- $V_1 + \cdots + V_t$ *is a* coloring *if V_i is an independent set,* $\forall i \in [t]$.
- $V_1 + \cdots + V_t$ *is a* clique cover *if V_i is an clique,* $\forall i \in [t]$.
- chromatic number $\chi(G) = \min\{t : \exists \textit{ coloring } V_1 + \cdots + V_t \textit{ of } G\}$.
- clique cover number $\kappa(G) = \min\{t : \exists \textit{ clique cover } V_1 + \cdots + V_t \textit{ of } G\}$.

Note that a single vertex v is a clique as well as an independent set, so we always have $1 \leq \alpha(G), \omega(G) \leq |V|$. Also note that $V_1 + \cdots + V_t$ with $|V_i| = 1, \forall i \in [t]$ is a coloring and a clique cover. Thus, we always have $1 \leq \chi(G), \kappa(G) \leq |V|$.

The following table tracks the four parameters across the five important graph families.

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

Consider the following notes and observations: We use the following terms interchangeably 2-colorable $\Leftrightarrow \chi(G) \leq 2 \Leftrightarrow$ bipartite. We can observe that ω, χ and α, κ often are the same or similar.

Our aim in this lecture will be a polynomial algorithm for all 4 parameters.

2.3 Perfect graphs

This section introduces perfect graphs, their defining properties and the two important structural results (Theorem 9 and Theorem 16).

We begin with an observation.

▶ **Observation 5.** *For every Graph G we have $\chi(G) \geq \omega(G)$ and $\kappa(G) \geq \alpha(G)$.*

Proof. If $I \subseteq V_G$ is independent and $C \subseteq V_G$ is a clique, then $|I \cap C| \leq 1$. Hence, for **any** coloring $V_1 + \cdots + V_t = V_G$ and any clique C, we have $|C \cap V_i| \leq 1$ for $i \in [t]$. If $|C| = \omega(G)$, then $t \geq |C|$. Thus, $\chi(G) \geq \omega(G)$.

Analogously, for **any** clique-cover $V_1 + \cdots + V_t = V_G$ and any independent set I, we have $|I \cap V_i| \leq 1$ for $i \in [t]$. If $|I| = \alpha(G)$, then $t \geq |I|$. Thus, $\kappa(G) \geq \alpha(G)$. ◀

The main question of AGT is when these inequalities turn into equalities. Here, the boring answer is that any graph may be modified to fulfil these equalities by adding a large clique or independent set. Due to this we are interested in the cases when the equalities hold for all induced subgraphs.

In the following we consider two exponential sets of restrictions.

▶ **Definition 6.** *Consider two properties.*

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

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

We begin by considering our important graph families. Here, we note that $K_n, E_n, K_{n,m} P_n$ and C_n for even n all fulfil (P1) and (P2), while C_n for odd n fulfil neither.

We also observe the following:

▶ **Observation 7.** *If $G + H$ are vertex-disjoint, $\alpha(G + H) = \alpha(G) + \alpha(H)$ and $\kappa(G + H) = \kappa(G) + \kappa(H)$.*

We are now ready to define perfect graphs.

▶ **Definition 8.** *A graph G is called* perfect*, if G has (P1) and (P2).*

We observe that C_5 has $\omega(C_5) = 2$, but $\chi(C_5) = 3$ and $\alpha(C_5) = 2$, but $\kappa(C_5) = 3$. Thus, C_5 is not perfect. Furthermore, we note that this is the smallest such graph.

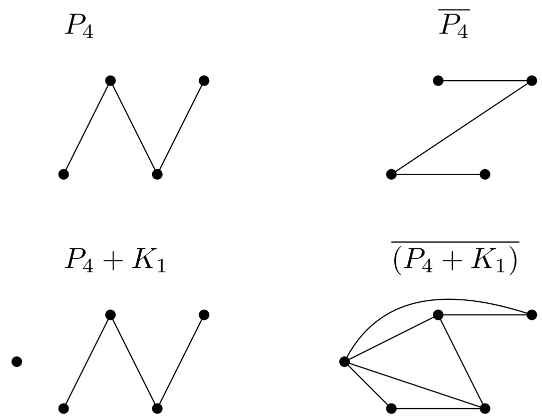
We continue by considering how (P1) and (P2) relate to each other.

▶ **Theorem 9** (Weak perfect graph theorem (WPGT))**.** *For every graph G it holds: G has (P1) $\Leftrightarrow$ G has (P2).*

Warning: $\forall A \subseteq V_G : \chi(G_A) = \omega(G_A) \Leftrightarrow \kappa(G_A) = \alpha(G_A)$ is not true. This is due to the fact that (P1) and (P2) may break on different subsets.

Before proving this theorem we consider a different approach of defining perfect graphs and stating the WPGT.

▶ **Definition 10.** *For graph $G = (V, E)$ the* complement *of G is the graph $\overline{G} = (V, \overline{E})$, where $\overline{E} = \binom{V}{2} - E$.*



We can observe the following relations:

graph G	A clique	$V_1 + \dots + V_t$ coloring
complement $\overline{G}$	A independent set	$V_1 + \dots + V_t$ clique cover

Thus, $\alpha(G) = \omega(\overline{G})$ and $\chi(G) = \kappa(\overline{G})$. Similarly, (P1) for $G \Leftrightarrow$ (P2) for $\overline{G}$ and (P2) for $G \Leftrightarrow$ (P1) for $\overline{G}$.[1]

To prove the WPGT we consider (P3) $\forall A \subseteq V_G : \omega(G_A) \cdot \alpha(G_A) \geq |A|$. This property connects ω and α and informally states that not both parameters can be small. Note that C_5 has $\alpha(C_5) \cdot \omega(C_5) = 2 \cdot 2 \ngeq 5 = |C_5|$ and thus fails (P3) in addition to (P1) and (P2).

We prove that G has (P1) $\Leftrightarrow$ G has (P2) $\Leftrightarrow$ G has (P3).

We first introduce a technique called vertex replication.

▶ **Definition 11.** *For a graph $G = (V, E)$ and $h \in \mathbb{N}^V$ we define $G \circ h$ as the graph on the vertex set $V(G \circ h) = \bigcup_{v \in V} \{v^1, \dots, v^{h(v)}\}$ and edges $u^i v^j$ if and only if $i \in [h(u)], j \in [h(v)], uv \in G$. This is called a* vertex replication *or a* vertex repetition *of G.*

Note that for us $\mathbb{N} = \{0, 1, 2, \dots\}$.

▶ **Definition 12.** *Let $\mathbb{1}$ be $\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix} \in \mathbb{N}^V$ and let $G = (V, E)$ be a graph. Let e_i be $\begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}$ the i-th unit vector in $\mathbb{N}^V$. For vertex $v \in V$ define $G \circ v$ as $G \circ h$ with $h(x) = \begin{cases} 1 & x \neq v \\ 2 & x = v \end{cases}$. So $h = \mathbb{1} + e_i$, if v is the i-th vtx. Define $G - v$ as $G \circ h$ with $h = \mathbb{1} - e_i$. These are called* elementary operations.

▶ **Observation 13.** *Every $G \circ h$ can be obtained from G by a sequence of elementary operations.*

We consider how vertex replication interrelates with our properties.

▶ **Lemma 14** (Lemma 2.6)**.** *For G and $H = G \circ h$, we have:*
i) (P1) for $G \Rightarrow$ (P1) for H.

---

[1] In literature sometimes perfect graphs are defined as fulfilling (P1). Then, the WPGT states that G perfect $\Leftrightarrow \overline{G}$ perfect.

ii) (P2) for $G \Rightarrow$ (P2) for H.

Proof. We consider the two statements separately.

i) We assume w.l.o.g $H = G \circ v$ or $H = G - v$. We now consider two cases:

- Case $H = G - v$: Then $H = G_{V-v}$ hence (P1) for $G \Rightarrow$ (P1) for H as H is a induced subgraph of G.

- Case $H = G \circ v$: Here, the vertex v is replaced by the vertices v^1, v^2. We note that $H - v^1 \cong H - v^2 \cong G$ and take $A \subseteq V_H$. If $|A \cap \{v^1, v^2\}| < 2$, then $A \subseteq V_G$, hence $\chi(H_A) = \chi(G_A) \overset{\text{(P1) on } G}{=} \omega(G_A) = \omega(H_A)$. Thus, let $v^1, v^2 \in A$ and consider $A' = A - v^1 \subseteq V_G$.

  By (P1) for G we have $\chi(G_{A'}) = \omega(G_{A'})$. Since we can modify this coloring by adding v^1 to the same color class as v^2, as the two vertices have the same neighbours but share no edge, we get $\chi(H_A) \leq \chi(G_{A'})$. As adding a vertex cannot decrease the clique number, we get $\omega(G_{A'}) \leq \omega(H_A)$. Using a previous observation (Observation 5) we can puzzle this together: $\chi(H_A) \leq \chi(G_{A'}) = \omega(G_{A'}) \leq \omega(H_A) \leq \chi(H_A)$. Since this chain of inequalities starts and ends with the same parameter, all inequalities must be equal. So we have (P1) for H.

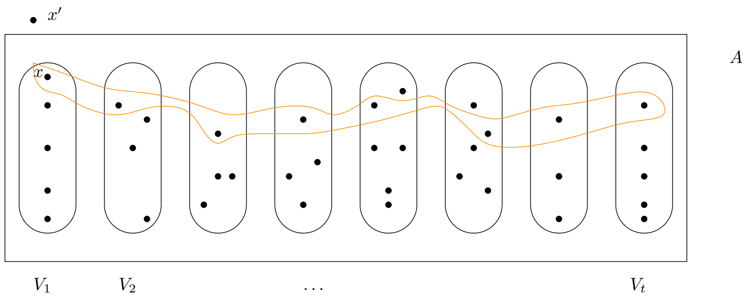
ii) Let G have (P2). We assume w.l.o.g. $H = G \circ x$ (or trivially $H = G - v$). Let x, x' be the two copies of x in H. As argued before we assume that w.l.o.g. A' contains x, x'. Let $A = A' - x' \subseteq V_G$. We note that (P2) for $G \Rightarrow \kappa(G_A) = \alpha(G_A) \Rightarrow V_1 + \cdots + V_t$ clique cover of $G_A = H_A$ with $t = \alpha(H_A)$. So every independent set I of H_A with $|I| = t$ contains one vertex per V_i. We now distinguish on whether x is in any such independent set.

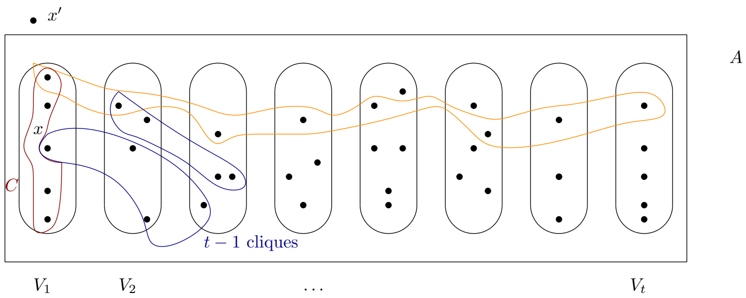
- Case 1: $\exists I \subseteq A$ independent set of H_A with $|I| = t, x \in I$, then $I + x'$ is independent set in $H_{A'}$. So $\alpha(H_{A'}) \geq t + 1$. We also note that $V_1 + \cdots + V_t + \{x'\}$ is a clique cover of $H_{A'}$. Thus, we have $\kappa(H_{A'}) \leq t + 1 \leq \alpha(H_{A'})$ using previous observations (Observation 5) we obtain equalities.

- Case 2: $\forall I \subseteq A$ i-set of H_A with $|I| = t : x \notin I$. Let $C = V_1 - x$ then H_{A-C} has $\alpha(H_{A-C}) \leq t - 1$. Due to (P2) for G we know $\exists$ clique cover $V_1' + \cdots + V_{t-1}'$ of $G_{A-C} = H_{A-C}$ with $\leq t - 1$ cliques. We construct a new clique cover and note that $V_1' + \cdots + V_{t-1}' + (C + x')$ is clique cover of $H_{A'}$. Thus, $\kappa(H_{A'}) \leq t \leq \alpha(H_{A'}) \leq \kappa(H_{A'})$. Again we have equality.

◀

The proof of ii) is visualized in the following graphic.





We use this result to prove a lemma needed for the WPGT.

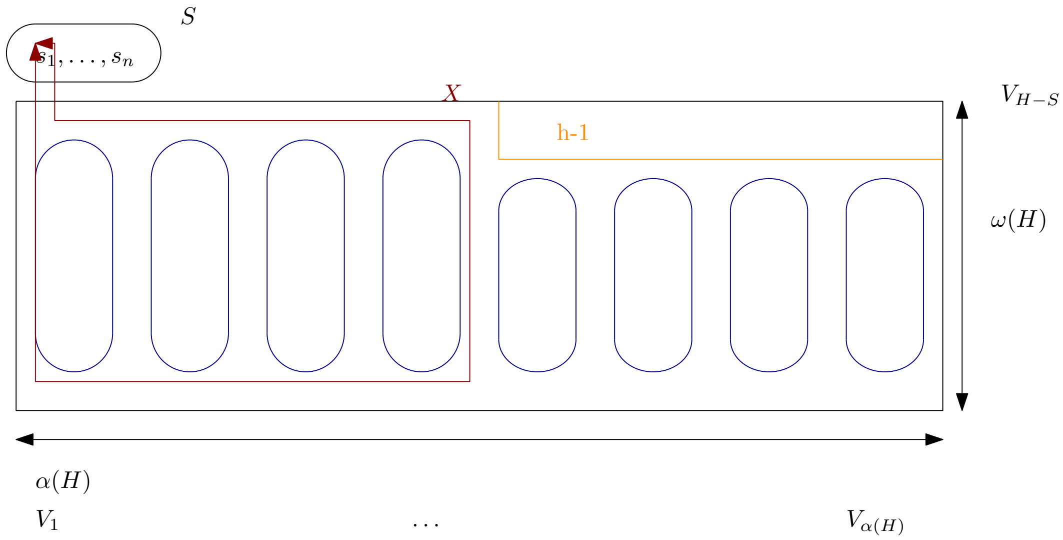
▶ **Lemma 15** (Lemma 2.7). *If $H = G \circ h$ then,*

$$\left.\begin{array}{l}\text{(P2) for all proper induced subgraphs of } G\\ \text{(P3) for } G\end{array}\right. \Rightarrow \text{(P3) for } H.$$

Proof. Assume f.s.o.c. that (P3) does not hold for H. We assume w.l.o.g. $\forall A \subseteq V_H, A \neq V_H : \omega(H_A) \cdot \alpha(H_A) \geq |A|$ but $\omega(H) \cdot \alpha(H) < |V_H|$. Otherwise we take a smaller H as counterexample. So some vertex s of G has $h(s) \geq 2$, since otherwise H is subset of G. So in H we have $S = \{s_1, \dots, s_h\}$. Consider $H - s_h$, this graph has (P3) per assumption. Thus, $|V_H| - 1 \leq \omega(H - s_h) \cdot \alpha(H - s_h) \leq \omega(H) \cdot \alpha(H) \leq |V_H| - 1$ using the above inequality. Again, we get a chain of equalities. Due to this we know $\omega(H) \cdot \alpha(H) = |V_H| - 1, \alpha(H - s_h) = \alpha(H), \omega(H - s_h) = \omega(H)$. By iteratively applying $\alpha(H) = \alpha(H - s_h)$ we get $\alpha(H - S) = \alpha(H)$.

As $H - S$ is obtained from $G - S$ by vertex multiplication and since $G - S$ has (P2) we know due to Lemma 2.6 (Lemma 14) that $H - S$ has (P2). Take a clique cover $V_1 + \dots + V_{\alpha(H)}$ of $H - S$. Then, we can use $|V_H - S| = V_H - h = \omega(H) \cdot \alpha(H) - (h - 1)$. Here, the minus one is due to the $V_H - 1$. Also $|S| = h \leq \alpha(H)$ since S is an independent set in H. As we have a clique cover of $\alpha(H)$ cliques in a graph of $\omega(H) \cdot \alpha(H)$ vertices, each clique -bar one - in the cover has size $\omega(H)$ before removing S. So at most $h - 1$ of $V_1, \dots, V_{\alpha(H)}$ have size $< \omega(H)$. We assume w.l.o.g. $|V_1| = \dots = |V_{\alpha(H)-(h-1)}| = \omega(H)$. Let $X = V_1 + \dots + V_{\alpha(H)-(h-1)} + s_1$. We can compute the size of X. $|X| = (\alpha(H) - (h - 1)) \cdot \omega(H) + 1$. Due to our definition of X we have $\omega(H_X) = \omega(H)$. Due to (P3) for H_X we have $\alpha(H_X) \geq \lceil \frac{|X|}{\omega(H_X)} \rceil = \lceil \frac{(\alpha(H)-(h-1)) \cdot \omega(H)+1}{\omega(H)} \rceil = \alpha(H) - (h - 1) + 1$. Here, we use the ceiling as we consider integer values and lower bounds. So $\exists I$ independent set in H_X, $|I| = \alpha(H) - (h - 1) + 1$, $s_1 \in I$. So $I + \{s_2, \dots, s_h\}$ is an independent set in $H \Rightarrow \alpha(H) \geq \alpha(H) + 1$ which is a contradiction. ◀

This proof is visualized below.



We can now prove the WPGT.

Proof. Let $G = (V, E)$ be a graph, we prove (P1)⇔(P2)⇔(P3) by induction on $|V|$. The base case of one vertex graphs is trivial.

- (P1)⇒(P3):
  Say (P1) holds for G. Let $A \subseteq G$. If $A \neq V_G$ then (P1) holds for G_A and by induction ⇒ (P3) holds for G_A, i.e. $\omega(G_A) \cdot \alpha(G_A) \geq |G_A|$. So we assume w.l.o.g. $A = V_G$, i.e. we need to show that $\omega(G) = \alpha(G) \geq |V_G|$. We know (P1)⇒ $\exists$ coloring $V_1 + \cdots + V_t = V_G$ with $t = \omega(G)$. Here, $|V_i| \leq \alpha(G), \forall i$. So $\omega(G) \cdot \alpha(G) \geq |V_G|$.
- (P3) ⇒ (P1):
  Let (P3) hold for G. To show (P1) it is enough (w.l.o.g) to show $\chi(G) \leq \omega(G)$. We consider all cliques of size $\omega(G)$.
  - Case 1: $\exists I$ independent set in G $\forall C$ clique, $|C| = \omega(G)$:$I \cap C \neq \emptyset$.
    We consider $G - I$ and note $\omega(G-I) \leq \omega(G) - 1$. So due to the induction hypothesis we have (P1) for (G-I), i.e. $V_1 + \cdots + V_t = V_G - I$ with $t \leq \omega(G) - 1$. Thus, $V_1 + \cdots + V_t + I$ is a coloring of G. So we have $\chi(G) \leq t + 1 = \omega(G) - 1 + 1$ and we are done.
  - Case 2: $\forall I$ i-set $\exists$clique $C(I)$, $|C(I)| = \omega(G), C(I) \cap I = \emptyset$:
    Consider the set of all independent sets $Y = \{I \subseteq V_G : I \text{ independent set}\}$. We choose $h(v) = \#\{I \in Y : v \in C(I)\}$ and consider $H = G \cdot h$. Since (P3) for G and (P2) for $G_A, A \subsetneq V_G$, Lemma 2.7 (Lemma 15) tells us that (P3) holds for H. Here, (P2) for all proper subgraphs holds due to induction.
    Say $V_H = X$. Then, $\omega(H) \cdot \alpha(H) \geq |V_H| = |X|$. We also know $|X| = \sum_{v \in V_G} h(v) = \omega(G) \cdot |Y|$. Also $\omega(H) \leq \omega(G)$ since each clique of H has at most one copy of each original vertex. We have $\alpha(H) = \max_{I \in Y} \sum_{v \in I} h(v) = \sum_{I' \in Y} |C(I') \cap I|$. Here each summand is 0 or 1. The second term is an alternate formulation of the sum where we sum over all other independent sets and consider how much they contributed to $h(v)$. This is $\leq |Y| - 1$ since $C(I) \cap I = \emptyset$. Combining this we have $\omega(G) \cdot (|Y| - 1) \geq \omega(H) \cdot \alpha(H) \geq |X| = \omega(G) \cdot |Y|$ which is a contradiction So case two does not happen.
- (P2)⇔(P3):
  We have (P2) for G ⇔ (P1) for $\overline{G}$ ⇔ (P3) for $\overline{G}$ ⇔ (P3) for G. In the last step we used that multiplication is commutative and that α and ω switch roles in the complement.

◀

To end this section we summarize our results:
So we know the following to be equivalent:

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

So far we know the following non-perfect graphs:

- odd cycle $C_t, t \geq 5$
- complements of C_t, odd $t \geq 5$
- every graph with induced odd C_t, odd $\overline{C_t}, t \geq 5$

It can be shown that these known non-perfect graphs are all that exist.

▶ **Theorem 16** (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 *perfect*

3 Intersection graphs

So far we have considered perfect graphs without further restrictions. This graph class is still to broad to find the desired polynomial algorithms for our four parameters. In this section we consider a subclasses of perfect graphs that are also intersection graphs.

▶ **Definition 17.** *A collection of sets $S = \{S(v) : v \in V\}$ is an* intersection representation *of $G = (V, E)$ if $uv \in \Leftrightarrow S(u) \cap S(v) \neq \emptyset$.*

3.1 Interval graphs

We begin by considering interval graphs which are a subclass of intersection graphs.

▶ **Definition 18.** *G is an* interval graph *if G has an intersection representation with intervals of $\mathbb{R}$, i.e. $I = \{I(v) : v \in V\}$, $\forall I(v) = [l_v, r_v]$. $uv \in E \Leftrightarrow I(u) \cap I(v) \neq \emptyset \Leftrightarrow \min\{r_u, r_v\} \geq \max\{l_u, l_v\}$.*

▶ **Definition 19.** *For graph G and integer $t \geq 4$ we define:*

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

Due to the SPGT we know that a graph being perfect is equivalent to there being no odd hole and no odd anti-hole.

To show that interval graphs are perfect we consider their relation to holes.

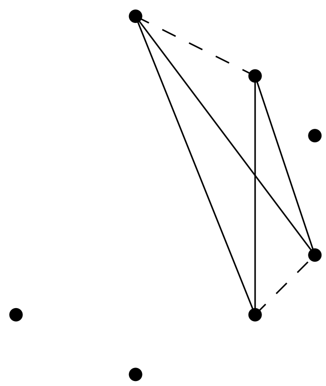
▶ **Lemma 20.** *G interval graph $\Rightarrow$ no t-hole for ≥ 4.*

Proof. Consider an interval representation $I = \{I(v) = [l_V, r_v] : v \in V\}$ and assume f.s.o.c. that there is a $t-$hole $C_t = [v_1, \ldots, v_t], t \geq 4$. Then, $I(v_{i-1}), I(v_{i+1})$ cover distinct endpoints of v_i. Thus, $I(v_1) \cap I(v_t) = \emptyset \Rightarrow v_1 v_t \neq E$. This is a contradiction. ◀

We use this result to prove perfectness.

▶ **Lemma 21.** *G interval graph $\Rightarrow$ G perfect*

Proof. We use the SPGT. We first note that G has no odd hole due to the previous lemma (Lemma 20). To show that G has no odd anti-hole we consider C_5 separately. Here, we have $\overline{C_5} = C_5$. For all other odd-anti-holes we find a 4-hole in them. Consider $\overline{C_t}, t \geq 7$:



So we find a 4-hole in $\overline{C_t}$, which cannot happen by the previous lemma. ◀

What we showed is actually: G interval graph $\Rightarrow$ G has no holes $\Rightarrow$ G is perfect.
Or more detailed: G interval graph $\Rightarrow$ G has no t-holes $t \geq 4$ $\Rightarrow$ G has no odd hole, has no odd anti-hole $\Rightarrow$ G is perfect. In the last step we used the SPGT.

In the next section we generalize these ideas.

3.2 Definition and recognition of chordal graphs

We begin by defining chordal graphs.

▶ **Definition 22.** *$G = (V, E)$ is* chordal*, if G has no t-hole, $t \geq 4$. Equivalently every, not necessarily induced, cycle $C_t, t \geq 4$ in G has a* chord*. Here, a chord is an edge uv with u, v non-consecutive on the cycle.*

We begin by considering examples of chordal graphs.

- complete graphs
- paths
- empty graphs
- trees, forests
- interval graphs
- more ...

Remember, that trees are very nice graphs because we can use divide and conquer to find fast algorithms. Furthermore, trees have leaves and thus we can induction-like build up trees.

We show that chordal graphs have similar vertices.

▶ **Definition 23.** *$G = (V, E)$ graph and vertex $v \in V$ is* simplicial *if Adj(v)=$\{u \in V : uv \in E\}$ is a clique.*

Our goal in the following is to show that every chordal graph has ≥ 1 simplicial vertex.

▶ **Lemma 24.** $\begin{matrix} v \text{ simplicial in } G \\ G - v \text{ perfect} \end{matrix} \Rightarrow G$ *perfect.*

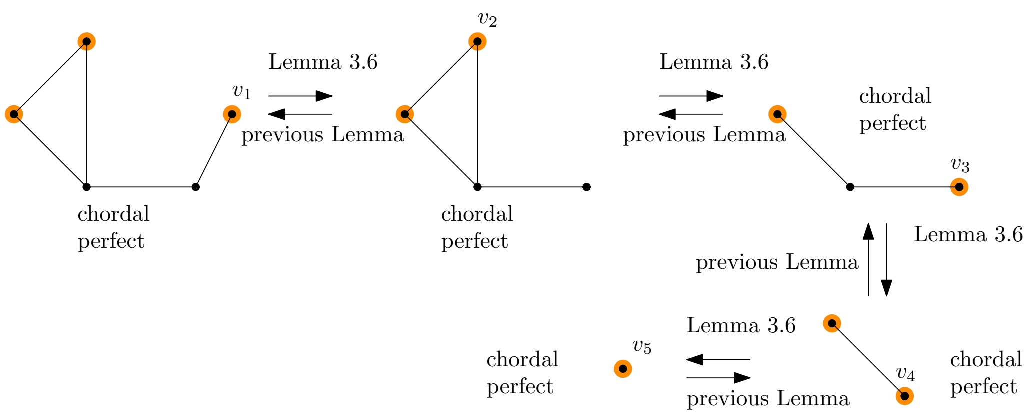
Proof. We verify (P1) $\forall A \subseteq V_G : \chi(G_A) = \omega(G_A)$. Consider any fixed $A \subseteq V_G$.

- Case $v \notin A$:
  Then, $A \subseteq V_{G-v}$ and $\chi(G_A) = \omega(G_A)$ as $G - V$ is perfect
- case $v \in A$:
  Let A' be $A - v \subseteq V_G - v$. Then, $\chi(G_{A'}) = \omega(G_{A'})$ due to (P1) for G. So there is a coloring $A' = V_1 + \cdots + V_t$ with $t = \omega(G_{A'})$. We consider two cases.
  - Case 1: $|\text{Adj}(v) \cap A'| < t = \omega(G_{A'})$.
    Then, $\exists i : V_i \cap (\text{Adj}(v) \cap A') = \emptyset$. We add v to this V_i to get $V_i' = V_i + v$. So $\chi(G_A) \leq \chi(G_{A'}) = \omega(G_{A'}) \leq \omega(G_A) \leq \chi(G_A)$ and thus all these are equal.
  - Case 2: $|\text{Adj}(v) \cap A'| \geq t = \omega(G_{A'})$.
    Then, due to the fact that the neighbourhood of v is a clique we have $|\text{Adj}(v) \cap A'| = t = \omega(G_{A'})$. So $(\text{Adj}(v) \cap A') + v$ is a clique in G_A of size $t + 1$. So $\omega(G_A) \geq t + 1 = \omega(G_{A'}) + 1 = \chi(G_{A'}) + 1 \geq \chi(G_A) \geq \omega(G_A)$. Here, the second to last inequality is due to the fact that $V_1 + \cdots + V_t + \{v\}$ is a coloring of G on $t + 1$ colors.

◀

In the following we want to remove such simplicial vertices iteratively. We thus must show that the class of chordal graphs is closed under the taking of subsets.

▶ **Observation 25.** *G chordal $\Rightarrow \forall A \subseteq V_G : G_A$ is chordal. In particular $G - v$ is chordal $\forall v \in V$.*



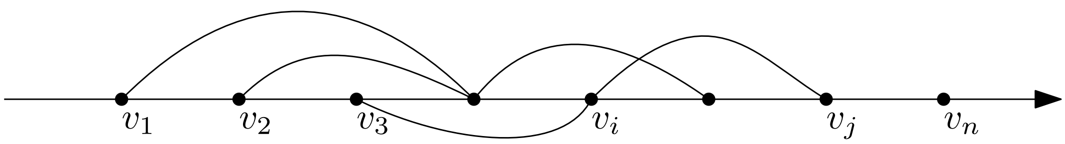
If each chordal graph has a simplicial vertex we can remove one such vertex in each step while maintaining chordality. We end with a K_1 which is trivially perfect. We then use the previous lemma (Lemma 24) to go back and maintain perfectness.

We formalize this idea:

▶ **Definition 26.** *For graph $G = (V, E), |V| = n$, a* perfect elimination sceme (PES) *of G is a vertex ordering $\sigma : [v_1, \ldots, v_n]$ s.t. v_i is simplicial in $G_{\{v_i, \ldots, v_n\}}, \forall i \in [n]$.*

So by the previous lemma (Lemma 24) we know that graphs with a PES are perfect.

We visualize a vertex ordering $\sigma : [v_1, \ldots, v_n]$ in the following fashion.



We then say that v_i is left/before of v_j in σ and that v_j is right/after v_i. If σ is a PES, then every right neighbourhood $\mathrm{Adj}(v_i) \cap \{v_i, \dots, v_n\}$ is a clique.

In the next step lay the groundwork to prove that each chordal graph has a simplicial vertex.

▶ **Definition 27.** *For a graph $G = (V, E)$, $S \subseteq V$ is a* separator *if $G - S$ is disconnected. If a, b are non-adjacent vertices in G, S is a* a,b-separator*, if a, b are in different components of $G - S$.*

Our goal is to find a separator S that is a clique in each chordal G that is not complete.

▶ **Lemma 28** (Lemma 3.4). $\begin{array}{l} G \, chordal, a, b \in V, ab \notin E, a \neq b \\ S \subseteq V_G is\ an\ inclusion\text{-}minimal\ a, b - separator \end{array} \Rightarrow S$ *is a clique.*

Proof. If $|S| \leq 1$, then S is a clique. So we assume $|S| \geq 2$. We take $x, y \in S, x \neq y$ and show that $xy \in E$. First, note that $S - x$ is not a a, b-separator. In the following we use G_A, G_B for the components of $G - S$ with $a \in A$ and $b \in B$. We know that x has an edge to A and to B (so does y). Consider the cycle $[x, a_1, \dots, a_p, y, b_1, \dots, b_q]$ and take C to be the shortest such cycle. Then, C has at least 4 vertices. Since G is chordal C has a chord e. Where is e?

- $e = a_i a_j$? No, as C is shortest
- $e = b_i b_j$? No, as C is shortest
- $e = a_i b_j$? No, as G_A, G_B are distinct components
- $e = x a_i$? No, as C is shortest
- $e = y a_i$? No, as C is shortest
- $e = x b_i$? No, as C is shortest
- $e = y b_i$? No, as C is shortest

So e must be $xy \in E$. ◀

We use this lemma to prove the desired result. As we use induction we show a stronger result.

▶ **Lemma 29** (Lemma 3.6). *Let G be chordal. Then,*

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

Proof. We use induction on $n = |V_G|$.
$n = 1$: $G = K_1$ and we are done.
For $n \geq 2$:
If $G \cong K_n$, then every vertex is simplicial. So we have $G \ncong K_n$. Let $a, b \in V_G, ab \notin E$ and let S be inclusion-minimal a, b-separator. In the following we consider the components of $G - S$. Here, G_A contains a and G_B contains b. Apply induction on G_{S+A} and G_{S+B}. These are smaller since a or b are missing and these are chordal. In G_{S+A} either all vertices are simplicial or there are two non-adjacent simplicial vertices, by induction. By Lemma 3.4 (Lemma 28) there is a simplicial vertex $x \in A$. This is due to the fact that either all vertices are simplicial and we can choose any vertex or that at most one non-adjacent vertex can be part of the clique S. This vertex is also simplicial in G as $\mathrm{Adj}_{G_{S+A}}(x) \subseteq S + A$ Using a symmetrical argument we get: $\exists y \in B$ simplicial in G. Since A and B are different components we have $xy \notin E_G$. ◀

We thus achieved our goal of showing that each chordal graph has a simplicial vertex.

Consider the following definitions:

(i) G chordal, i.e. every cycle of length ≥ 4 has a chord
(ii) every induced cycle is a triangle (no-t-hole)
(iii) every inclusion-minimal separator is a clique
(iv) every induced subgraph has a simplicial vtx
(v) there is a PES

So far we have seen the equivalency of (i) and (ii) as well as the implications (ii)⇒(iii) by Lemma 3.4 (Lemma 28), (iii)⇒(iv) by Lemma 3.6 (Lemma 29) and (iv)⇒(v). By showing (v)⇒ (i) we show that all these definitions are equivalent.

Proof. Let G be a graph, σ as PES and C cycle of length ≥ 4 in G. Also let v be the leftmost vertex of C in σ, say $v = \sigma(i)$. Consider $x, y \in \mathrm{Adj}(v) \cap \{\sigma(i+1), \ldots, \sigma(n)\}$. Since σ is a PES we have $xy \in E_G$. So any vertices x, y on C that are right of v must share an edge which is a chord. ◀

So (v) leads to a trivial recognition algorithm with runtime $\mathcal{O}(n^4)$ as we need to find a simplicial vertex n times.

Consider the following algorithm called *LexBFS*.

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

1 assign each vertex label ∅;
2 for i ← n to 1 do
3 |  choose a vertex v
|        with no assigned number in σ
|            with lexicographically largest label;
4 |  σ(i) ← v;
5 |  for every vertex w ∈ Adj(v)
|  |       with no assign number in σ
6 |  |  append i to label(w);
7 |  end for
8 end for
```

Algorithm 1 : LexBFS

We use this algorithm to build a simple recognition algorithm for chordal graphs based on property (v). We use LexBFS to find a vertex ordering σ that is a PES if and only if the graph was chordal.

Here, we have two viewpoints of LexBFS.

- Viewpoint 1: We have labels at each vertex and consider strings over the alphabet $\{1, \ldots, n\}$. We use the lexicographical order $1 <_{lex} \cdots <_{lex} n$. So for label($v$)=$\alpha_1 \ldots \alpha_s$ and label(u)=$\beta_1 \ldots \beta_t$ we have $\alpha = \alpha_1 \ldots \alpha_s <_{lex} \beta_1 \ldots \beta_t = \beta \begin{cases} \alpha_1 <_{lex} \beta_1 \\ \alpha = \emptyset, \beta \neq \emptyset \\ \alpha_1 = \beta_1 \text{ and } \alpha_2 \ldots \alpha_s <_{lex} \beta_2 \ldots \beta_t \end{cases}$
- Viewpoint 2: We consider a queue of all not numbered vertices with $v \in$First(Q). The elements of Q are sets of vertices of the same label, sorted lexicographically in Q. Then for Adj(v) we split each set X in Q into Adj(v) $\cap X$ and $X-$Adj(v).

The plan in the following is to run LexBFS and obtain a vertex-ordering σ and then test if σ is PES in linear time. For this we must prove σ PES$\Leftrightarrow$ G chordal and implement LexBFS in linear time.

We use the following lemma to characterize LexBFS results.

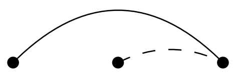
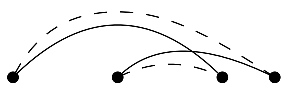
▶ **Lemma 30.** *$\sigma \in$ LexBFS(G), then $\forall a, b, c \in V_a$ it holds $a <_\sigma b <_\sigma c$ and $ac \in E_G, bc \notin E_G \Rightarrow \exists d$ with $c <_\sigma d$ and $ad \notin E_G, bd \in E_G$.*

Proof. Consider such a triplet. When c is processed by LexBFS one of the following cases occurs.

- If label(a)=label(b), then afterwards label(a)$>_{lex}$label(b) and thus this will still hold when b is processed, contradicting the choice of b.
- If label(a)$\neq$label(b), then label(b)$<_{lex}$label(a). Consider the step before the first time, when label(a)$\neq$label(b). This occurs when processing vertex d, $c <_\sigma d$. It holds that $b \in$ Adj(d) and $a \notin$ Adj(d) as $a <_\sigma b$. So we have $bd \in E, ad \notin E$.

◀

Note

If I have  , I can conclude, that 

We use this property to show the desired result.

▶ **Theorem 31** (Theorem 3.9). *G is chordal if and only if LexBFS outputs a PES*

Proof. '$\Leftarrow$' clear
'$\Rightarrow$' We prove the contraposition, i.e. σ not PES $\Rightarrow$ G not chordal. Consider a σ not PES, then $\exists a, b, c; a <_\sigma b <_\sigma c; ab, ac \in E_G; bc \notin E_G$. We chose a triplet with maximally right c. We use the naming convention $a = x_0, b = x_1, c = x_2$. By the Lemma 30 we know: $\exists x_3 : x_2 <_\sigma x_3; x_1x_3 \in E_G; x_0x_3 \notin E_G$. We consider two cases:

i) $x_2x_3 \in E_G$:
Then, $x_0x_1x_2x_3$ induces a C_4 and G is not chordal.

ii) $x_2x_3 \notin E_G$:
By Lemma 30 we know: $\exists x_4 x_2x_4 \in E_G, x_1x_4 \notin E_G, x_3 <_\sigma x_4$. If $x_0x_4 \in E_G$, the choice of x_2 as rightmost is contradicted. So we have $x_0x_4 \notin E_G$: If $x_3x_4 \in E_G$:, then we find $G[x_0, \ldots, x_4] = C_5$. If $x_3x_4 \notin E_G$, then we find $G[x_0, \ldots, x_4] = P_5$ with endpoints x_3, x_4. We continue and get $\exists x_5$ by Lemma 30. If $x_0x_5 \in E_G$, then $x_0x_2x_5$ forms a PES-triple with x_5 further to the right. This is a contradiction. So $x_0x_5 \notin E_G$. If $x_1x_5 \in E_G$, then we get a contradiction to the choice of x_3. Similarly, $x_4x_5 \in E_G$ implies an induced C_6 on $x_0 \ldots x_5$. And, $x_4x_5 \notin E_G$ implies an induced P_6 and the argument continues.

Since the graph is finite we eventually find the desired induced cycle. ◀

In the next step we want to show how LexBFS can be implemented in linear time.

LexBFS in $\mathcal{O}(|V| + |E|)$:
We use the following datastructure: We use a queue Q with sets that supports First(Q) and is implemented as a double-linked list. For each set S of vertices in Q we use a non-empty doubly-linked list and a Flag(S) that is true if S has been split. For each vertex w we store the set $S(w)$ that includes w. Finally, we need a fixlist L which is a list of all sets, that have been split.

We then use the following algorithm for the update step.

1 **for** $w \in \mathrm{Adj}(v)$ not numbered **do**
2   **if** $\mathrm{Flag}(\mathrm{Set}(w)) = \mathrm{false}$ **then**
3   insert new set S before $\mathrm{Set}(w)$ into Q;
4     $\mathrm{Flag}(\mathrm{Set}(w)) \leftarrow \mathrm{true}$; add $\mathrm{Set}(w)$ to $\mathrm{FixList}$;
5   **end if**
6   $S \leftarrow$ set before $\mathrm{Set}(w)$ in Q;
7   remove w from $\mathrm{Set}(w)$; add w to S;
8   $\mathrm{Set}(w) \leftarrow S$;
9 **end for**
10 **for** $S \in \mathrm{FixList}$ **do**
11   $\mathrm{Flag}(S) \leftarrow \mathrm{false}$;
12   **if** S empty **then**
13   remove S from Q;
14   **end if**
15   remove S from $\mathrm{FixList}$;
16 **end for**

Algorithm 2 : Update step in LexBFS

We then use the following runtime analysis: Line 1 to 9 is linear in $|\mathrm{Adj}(v)|$ and line 10 to 16 is linear in $|\mathrm{FixList}| = |\mathrm{Adj}(v)|$. So the update step can be done in $|\mathrm{Adj}(v)|$. Thus, the total runtime of LexBFS is $\mathcal{O}(\sum_v |\mathrm{Adj}(v)| + |V|) = \mathcal{O}(|V| + |E|)$.

It remains to test, if the output of LexBFS is PES.

The naive approach for such a test would be to test all triplets for the property. This takes $\Theta(n^3)$. Alternatively one may test the right neighbourhood of each vertex for cliques. This takes $\sum_v |\mathrm{Adj}(v)|^2 \approx \mathcal{O}(n^3)$. This approach looks at vertices more than once, so there is potential for improvement.

The idea is for v to tell its leftmost right neighbour u a set of vertices that should be pairwise adjacent. These form a clique. The vertex v also wants u to be adjacent to all of those.

We use the following algorithm.

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

▶ **Theorem 32.** *Algorithm 3 is correct.*

Proof. We must show: Algo 3 returns true $\Leftrightarrow$ σ is PES of G.
Equivalently we can show: Algo 3 returns false $\Leftrightarrow$ σ is not PES of G.
'$\Rightarrow$' $\exists$ vtx u with $A(u) - \text{Adj}(u) \neq \emptyset$, say $w \in A(u) - \text{Adj}(u)$. Who put $w \in A(u)$? This was done by some v earlier. So u is leftmost in $X_v, w \in X_v - u$. We thus found a tripled forbidden by PES and the result is no PES.

'$\Leftarrow$' Assume σ is not PES and take a forbidden triplet u, v, w with u, v closest together. We claim that u is the leftmost right neighbour of v. To show this we consider a vertex a inbetween: Consider $a \in X_v$, $v < a < u$. If $au \notin E_G$ the choice of the triple is contradicted as vau can be used. So $au \in E_G$. If $aw \notin E_G$ the choice of the triple is contradicted as vaw can be used. So $aw \in E_G$, but then auw is a better triple. So u is the leftmost in X_v.

So Algo 3 puts w into $A(u)$ when processing v. Later when processing u we have $w \in A(u) - \text{Adj}(u)$ and return false. ◀

Next we consider the runtime of this algorithm.

▶ **Theorem 33.** *Algo 3 can be done in $\mathcal{O}(|V| + |E|)$.*

Proof. We for-loop over each vertex once. Here, lines 2 to 7 are possible in $|\text{Adj}(v)|$. Line 7 appends $X - u$ to $A(w)$ without checking for duplicates. So this takes $\mathcal{O}(\sum_v |\text{Adj}(v)|) = \mathcal{O}(|V| + |E|)$. The check in line 8 to 10 uses the below algorithm. Here the test runs in $\mathcal{O}(|A(v)| + |\text{Adj}(v)|)$. This is also in $\mathcal{O}(|V| + |E|)$ since this list cannot be longer than the time spend to build it up. ◀

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

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

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

So we can recognize in linear time whether G is chordal and compute a PES of G.

3.3 Algorithms on chordal graphs

The aim of this section is to compute $\chi(G), \omega(G), \alpha(G)$ and $\kappa(G)$ for chordal graphs using a PES σ.

- Algo 5 finds $\omega(G)$ and $\chi(G)$ with clique C and coloring Φ optimal
- Algo 6 finds $\alpha(G)$ and $\kappa(G)$ with independent set U and clique cover Ψ optimal

Note that previously we defined a coloring as $V_1 + \cdots + V_n$ where V_i is an i-set. Equivalently we can use $\Phi : V \to [t]$ with $\Phi(v) = i \Leftrightarrow v \in V_i$ and $\Phi(v) = 0$ for uncolored vertex.

```
Input  : chordal graph G = (V, E).
Output : clique C and coloring φ.

1  compute with LexBFS a PES σ of G;
2  C ← ∅, φ ← 0;
3  for i ← n to 1 do
4  |  v ← σ(i);
5  |  X_v ← Adj(v) ∩ {σ(i+1), ..., σ(n)};
6  |  φ(v) ← min(ℕ − {φ(w) | w ∈ X_v});
7  |  if |C| < |X_v + {v}| then
8  |  |  C ← X_v + {v};
9  |  end if
10 end for
11 return C and φ;
```

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

▶ **Theorem 34.** *Algo 5 computes a clique C and a coloring Φ with $|C| = \omega(G)$ and $\max_v \Phi(v) = \chi(G)$.*

Note that we traverse the PES from right to left.

Proof. We show the different partial statements.

- C is a clique:
  Note that C is of the form $X_v + v$. As σ is a PES we know that X_v is a clique. So $C = X_v + v$ is clique. We thus have $\max_v(|X_v| + 1) = |C| \leq \omega(G)$.
- Φ is coloring::
  We set the color $\Phi(v)$ of each vertex once and never change it, so $\Phi(v) \geq 1$. Let $uv \in E_G$. We can assume w.l.o.g. $u \in X_v$. Then, we choose $\Phi(v)$ to be different from $\Phi(u)$. So we obtain a coloring and we have $\chi(G) \leq \max_v \Phi(v)$.
- C and Φ are optimal:
  For every vertex v we have $\Phi(v) \leq |X_v| + 1$ as at most $|X_v|$ colors are blocked. Hence $\chi(G) \leq \max_v \Phi(v) \leq \max_v |X_v| + 1 = |C| \leq \omega(G) \leq \chi(G)$. Again the last inequality holds for all graphs. So we have equalities everywhere and thus $|C| = \omega(G)$ and $\max_v \Phi(v) = \chi(G)$.

◀

We consider the runtime.

▶ **Theorem 35.** *Algo 5 can be done in $\mathcal{O}(|V| + |E|)$.*

Proof. The for-loop iteration for vertex v takes $\mathcal{O}(|\mathrm{Adj}(v)|)$. Here, line 6 is similarly to Algo 4 doable in $\mathcal{O}(|X_v|)$. Thus, the runtime is $\mathcal{O}(|V| + |E|)$. ◀

Input : chordal graph $G = (V, E)$.
Output : independent set U and clique cover ψ.

1 compute with LexBFS a PES σ of G;
2 $U \leftarrow \emptyset$, $\psi \leftarrow 0$;
3 **for** $i \leftarrow 1$ **to** n **do**
4     $v \leftarrow \sigma(i)$, $X_v \leftarrow \mathrm{Adj}(v) \cap \{\sigma(i+1), \ldots, \sigma(n)\}$;
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 ψ;

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

▶ **Theorem 36.** *Algo 6 computes a independent set U and a clique cover Ψ with $|U| = \alpha(G)$ and $\max_v \Psi(v) = \kappa(G)$.*

Note that we traverse the PES from left to right.

Proof. We show the different partial statements.

- U is an independent set:
  We use the following invariant: $w \in U, v >_\sigma w, \Psi(w) = 0 \Rightarrow vw \notin E_G$ equivalently $w \in U : v >_\sigma w : vw \in E_G \Rightarrow \Psi(v) = 0$. This invariant is true since $v \in X_w$ gets assigned $\Psi(v) \leftarrow |U| \neq 0$. So we have $|U| \leq \omega(G)$.
- Ψ is a clique cover
  In line 8 we set $\Psi(w) \leftarrow |U| = i, \forall w \in X_v + v$. Since σ is a PES, $X_v + v$ is a clique. Additionally, the value $|U|$ is never assigned again. In the final Ψ we have $\{v : \Psi(v) = i\} \subseteq X_v + v$ and thus this set is a clique.
- U and Ψ are optimal:
  We have $\kappa(v) \leq \max_v \Psi(v) = |U| \leq \alpha(G) \leq \kappa(G)$. Again the last step is true for all graphs. So we have equalities everywhere. I.e. $|U| = \alpha(G)$ and $\max_v \Psi(v) = \kappa(G)$.

◀

We consider the runtime.

▶ **Theorem 37.** *Algo 6 can be done in $\mathcal{O}(|V| + |E|)$.*

Proof. Similar to proof for Algo 5. ◀

3.4 On the relation between between intersection graphs and chordal graphs

In this section we aim to find an intersection representation of chordal graphs. We consider the following representation of substrees of a tree.

▶ **Definition 38.** *Let $G = (V_G, E_G)$ be a graph. We find a underlying tree $T = (V_T, E_T)$ such that we can assign each vertex of G a subtree T_v of T. We call the tree a* intersection representation as subtrees of a tree, *when $uv \in E_G$ if and only if $T_v \cap T_u \neq \emptyset$.*

The plan is to show that a graph has such an representation if and only if it is chordal. For this remember that interval graphs have a intersection representation of subtrees of a path.

The main ingredient we use in our proof is the Helly-property.

▶ **Definition 39.** *A family* $\{A_i\}_{i\in I}$ *of sets has the* Helly property*, if* $\forall J \subseteq I : \forall i, j \in J : A_i \cap A_j \Rightarrow \bigcap_{j\in J} A_j \neq \emptyset$.

So informally, this property requires pairwise intersection to imply intersection in one element.

The following proposition was proved in the exercises.

▶ **Proposition 40** (Proposition 3.13)**.** T *tree* $\Rightarrow \{T_i \subseteq T | T_i$ *subtree*$\}$ *has the Helly property.*

We use this in our main theorem:

▶ **Theorem 41.** *For every graph* $G = (V, E)$ *the following are equivalent:*

(i) G *is chordal*

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

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

Proof. We show the three implications and close a cycle.

- (ii) $\Rightarrow$ (i):
  Let G be a intersection graph of subtrees of a tree. Let $C = [v_1, \dots, v_k], k \geq 4$ be a cycle in G. We consider three subtrees of T. $T_1 = T_{v_1} \cup T_{v_2}, T_2 = T_{v_3} \cup \dots \cup T_{v_{k-1}}$ and $T_3 = T_{v_4} \cup \dots \cup T_{v_k}$ are subtrees as the trees of adjacent vertices are non distinct. We note that $T_1 \cap T_2 \neq \emptyset$ as $v_2v_3 \in E_G$, $T_2 \cap T_3 \neq \emptyset$ as $v_3v_4 \in E_G$ and $T_1 \cap T_3 \neq \emptyset$ as $v_kv_1 \in E_G$. So using the Helly-Property and Proposition 40 we get $\exists x \in V_T : x \in T_1, x \in T_2, x \in T_3$.
  We distinguish two cases:
  - Case 1: $x \in T_{v_1}$:
    Then, x is contained in $T_{v_j} \subseteq T_2$ for $j \in \{3, \dots, k-1\}$. So there is a chord.
  - Case 2: $x \in T_{v_2}$:
    Then, x is contained in $T_{v_j} \subseteq T_2$ for $j \in \{4, \dots, k\}$. So there is a chord.
- (i) $\Rightarrow$ (iii):
  Let $G = (V, E)$ be chordal. We use the notation $K(G) = \{X \subseteq V | X$ inclusion-maximal clique in $G\}$. We construct a tree and check for (*) $\forall v \in V, K_v = \{X \in K(G) | v \in X\} \subseteq K(G)$ induces a subtree in T. We find the tree T by induction on $|V|$. In the base case we have one vertex in G and one in $K(G) = T$. We can verify that (*) holds.
  In the induction step we consider $|V| \geq 2$. Let v be a simplicial vertex. By applying the induction hypothesis to $G - v$ we get a tree T' of $K(G - v)$.
  - Case 1: $\text{Adj}(v) \in K(G - v)$:
    Then, $\text{Adj}(v) + \{v\} \in K(G)$ and $K(G - v) - \text{Adj}(v) = K(G) - (\text{Adj}(v) + \{v\})$. We relabel the vertex in T' and get the new tree. We observe that (*) still holds as $\forall w \neq v$ nothing changes and v is only in one vertex label.
  - Case 2:$\text{Adj}(v) \neq K(G - v)$:
    Let $X \in K(G - v)$, $\text{Adj}(v) \subsetneq X$. Then, there is a vertex for X in T'. We add a new vertex $\text{Adj}(v) + \{v\}$ that is adjacent only to X. Then, (*) holds as $\forall w \in \text{Adj}(v) : w \in X$.
- (iii) $\Rightarrow$ (ii):
  Let $T = (V_T, E_T)$ be the tree with (*). Then, take T_v as the subtree induced by K_v. We

verify:

$$\begin{aligned} vw \in E_G &\Leftrightarrow \exists X \in K(G) : \{u, v\} \subseteq X \\ &\Leftrightarrow \exists X \in K(G) : X \in K_v, X \in K_w \\ &\Leftrightarrow \exists X \in V_T : X \in T_v \cap T_w \\ &\Leftrightarrow T_v \cap T_w \neq \emptyset \end{aligned}$$

Here, the first equality holds as $\{v, w\}$ is a clique.

◀

4 Comparability graphs

In this section we consider graphs where the vertices are given by elements and the edges by a better-than relation. So we consider directed edges (u, v) where v is better than u.

Formally, we use a binary relation.

▶ **Definition 42.** *A binary relation $\prec \subseteq V_G \times V_G = \{(u, v) : u \in V_G, v \in V_G\}$ is called* irreflexive*, if $v \not\prec v, \forall v \in V_G$, and* transitive*, if $\forall u, v, w : u \prec v \wedge v \prec w \Rightarrow u \prec w$. We call a irreflexive and transitive binary relation a* strict partial order*.*

Throughout this section we use the following notation: We consider only directed edges and have graphs $G = (V, E)$ wit finite V and $E \subseteq \{(u, v) : u, v \in V, u \neq v\} = V \times V - \{(w, w) : w \in W\}$. We again use the shorthand uv for (u, v), but note that now $uv \neq vu$. We call a graph $G = (V, E)$ *undirected*, if $\forall u \neq v : uv \in E \Rightarrow vu \in E$.

We begin this section by considering how we can orient such undirected graphs.

▶ **Definition 43.** *An* orientation *of a graph $G = (V, E)$ is $F \subseteq E$ such that $\forall uv \in E : uv \in F \Leftrightarrow vu \neq F$.*

▶ **Definition 44.** *For a subset $F \subseteq E$ we define $F^{-1} = \{vu : uv \in F\}$ to be the* reversal *of F. We also define $\hat{F} = F \cup F^{-1} = \{uv : uv \in F \text{ or } vu \in F\}$ to be the* (symmetric) closure *of F.*

We use this idea of orientations to define comparability graphs.

▶ **Definition 45.** *For an undirected graph $G = (V, E)$ an orientation F is called* transitive*, if $\forall a, b, c : ab \in F \wedge bc \in F \Rightarrow ac \in F$.*

▶ **Definition 46.** *A undirected graph $G = (V, E)$ is a* comparability graph*, if it admits a transitive orientation F. We then call G* transitively orientable*.*

We note that for example complete graphs and paths are comparability graphs.

▶ **Observation 47.** *F is transitive orientation $\Leftrightarrow F^{-1}$ is transitive orientation*

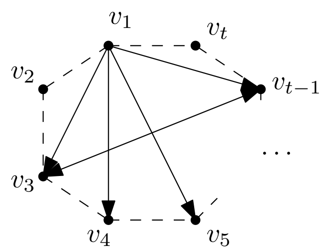
We now show that comparability graphs are perfect using the SPGT. This will also be implied by later structural results.

▶ **Theorem 48.** *G comparability graph $\Rightarrow G$ perfect*

Proof. We use the SPGT and first observe that if G is a comparability graph, then any subgraph $G_A, A \subseteq V_G$, also is a comparability graph. Hence it suffices to show that C_t and $\overline{C_t}$ are not comparability graphs for odd $t \geq 5$. Here, we take a transitive orientation F and show a contradiction.

We begin with odd cycles. W.l.o.g. we may assume $v_1v_2 \in F$. Using the transitivity of F we can conclude that $v_2v_3 \notin F \Rightarrow v_3v_2 \in F$ and $v_4v_3 \notin F \Rightarrow v_3v_4 \in F$. In general we know that each v_i with even i must be a sink and each v_i with odd i must be a source. Then, $v_tv_1 \in F, v_1v_2 \in F$ but $v_tv_2 \notin F$, so F is not transitive.

Next, we consider complements of odd cycles. We first note $\overline{C_5} = C_5$, so this case has already been handled. For $t \geq 7$, we may assume w.l.o.g. that $v_1v_3 \in F$. Since $v_4v_3 \notin E$, we have $v_1v_4 \in F$ or F not being transitive. So v_1 must be a source as this can be repeated for the other vertices. Using a symmetric argument, v_3 to v_{t-1} must be sinks. This yields a contradiction as this forces $v_3v_{t-1} \in F$ and $v_{t-1}v_3 \in F$.



◀

We note that C_t is a comparability graph for even t or $t = 3$. More general we note that all bipartite graphs are comparability graphs. Here, we use the orientation that orients each edge from the left set to the right.

We observe that the above proof used the following attribute: If $F \subseteq E$ is a transitive orientation and $ab \in F$ and $a'b' \in E$ where either $a = a'$ and $bb' \notin E$ or $a = a'$ and $aa' \notin E$, then $a'b' \in F$.

We formalize this notion.

▶ **Definition 49.** *We define the* Gamma-relation *as follows: For $ab \in E, a'b' \in E$ define $ab\Gamma a'b'$ if $a = a'$ and $bb' \notin E$ or $b = b'$ and $aa' \notin E$.*

We can restate our observation using this relation.

▶ **Observation 50.**
$$\begin{matrix} F \text{ transitive} & \\ ab \in F & \Rightarrow a'b' \in F. \\ ab\Gamma a'b' & \end{matrix}$$

We say that ab enforces or implies $a'b'$.

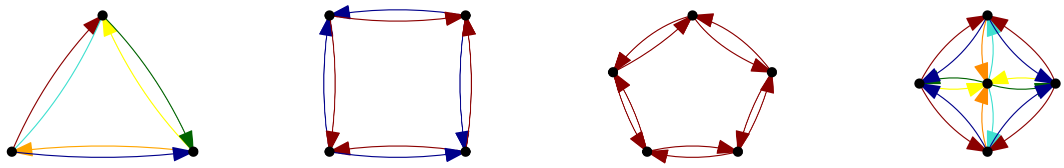
We now apply this result iteratively.

▶ **Definition 51.** *A* Γ-chain *is a sequence $a_1b_1, \ldots, a_kb_k$ of edges with not-necessarily distinct vertices such that $a_ib_i\Gamma a_{i+1}b_{i+1}, \forall i = 1 \ldots k$. We use $a_1b_1\Gamma^*a_kb_k$, where Γ^* is the transitive closure of Γ.*

We again restate our observation.

▶ **Observation 52.**
$$\begin{matrix} F \text{ transitive} & \\ ab \in F & \Rightarrow a'b' \in F. \\ ab\Gamma^* a'b' & \end{matrix}$$

We can see Γ^* as an equivalence relation of E as it is symmetric, transitive and reflexive. Here, symmetry follows from the two symmetric cases in the definition of Γ. Thus, Γ^* splits E into equivalence classes $\mathcal{I}(G)$. These are called *implication classes*.



$|\mathcal{I}(G)| = 6$ $|\hat{\mathcal{I}}(G)| = 3$ | $|\mathcal{I}(G)| = 2$ $|\hat{\mathcal{I}}(G)| = 1$ | $|\mathcal{I}(G)| = 1$ $|\hat{\mathcal{I}}(G)| = 1$

▶ **Observation 53.** *G comparability graph $\Rightarrow$ number of $\mathcal{I}(G)$ even.*

This is due to the fact that if there is a $A \in \mathcal{I}(G)$ with $ab, ba \in A$ then G is no comparability graph. The reverse implication is also true but non-trivial. We show this in the following.

▶ **Definition 54.** *For $A \in \mathcal{I}(G)$ we call $\hat{A} = A \cup A^{-1}$ a* color class *of G. We then define $\hat{\mathcal{I}}(G) = \{\hat{A} : A \in \mathcal{I}(G)\}$.*

Since $ab\Gamma^* a'b' \Leftrightarrow ba\Gamma^* b'a'$ we observe:

▶ **Observation 55.** *$ab\Gamma^* cd \Leftrightarrow cd\Gamma^* ab \Leftrightarrow ba\Gamma^* dc$*
$A \in \mathcal{I}(G) \Leftrightarrow A^{-1} \in \mathcal{I}(G)$

This results in the following theorem for transitive orientations.

▶ **Theorem 56** (Theorem 4.1)**.** *For $A \in \mathcal{I}(G)$ and transitive orientation F of G, we have $F \cap \hat{A} = A$ or $F \cap \hat{A} = A^{-1}$.*

Proof. We consider an edge $ab \in \hat{A}$. We assume w.l.o.g. $ab \in A$. This is valid due to Observation 55. We consider two cases:

- Case 1: $ab \in F$
  Then, we have $ab \in F \cap \hat{A}$. We take an edge $cd \in A$ with $ab\Gamma^* cd$. Due to Observation 52 we can follow the Γ-relations and get $cd \in F$. Hence, $A \subseteq F$. Since F is an orientation we have $F \cap F^{-1} = \emptyset$. So $A^{-1} \cap F = \emptyset$ and thus $F \cap \hat{A} = A$.
- Case 2: $ba \in F, ba \in A^{-1}$
  Here, we can apply an analogue argument.

◀

This theorem yields the following corollary.

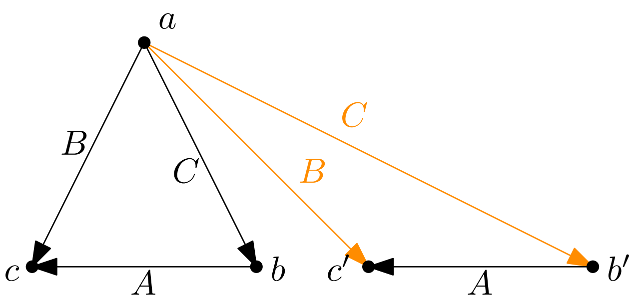
▶ **Corollary 57.** *For a comparability graph G and an implication class $A \in \mathcal{I}$ we have $A \cap A^{-1} = \emptyset$ and not $A = A^{-1}$.*

Proof. We consider such a graph and take a transitive orientation F. Then, $F \cap \hat{A} = A$ or $F \cap \hat{A} = A^{-1}$. But $F \cap F^{-1} = \emptyset$ and thus $A \cap A^{-1} = \emptyset$. ◀

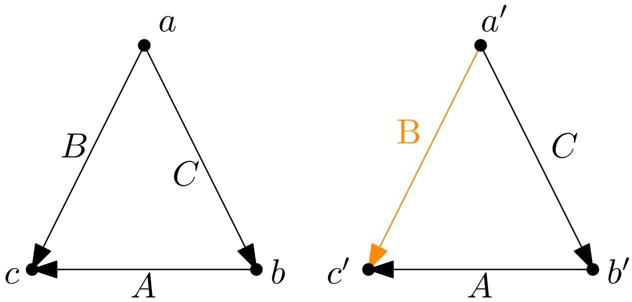
This corollary yields the first direction of the theorem used in our recognition algorithm. In the following we prove some preliminary results and then combine them to show the backwards direction.

▶ **Lemma 58** (Triangle-Lemma)**.** *For an undirected graph G, implication classes $A, B, C \in \mathcal{I}$ with $A \neq B, A \neq C^{-1}$ and a triangle abc in G the following two parts hold.*

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



(ii) $a'b \in C, b'c' \in A \Rightarrow a'c' \in B$

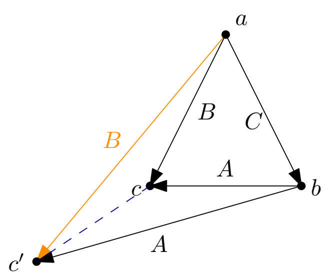


Here, the existence of the black edges, vertices and classes implies the orange ones. It is important to note that $A = C, B^{-1} = C, \ldots$ as well as $a' = b, b' = c, \ldots$ is possible.

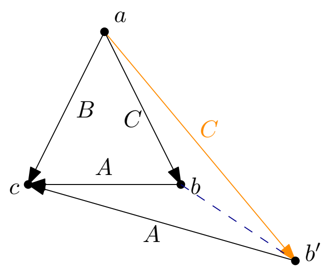
Proof. We prove the two parts separately.

(i) We first note that is enough to consider one step in the Γ-chain $bc\Gamma^* b'c'$. Here, two cases arise. Either $b = b', cc' \notin E$ or $c = c', bb' \notin E$.

- Case $b = b', cc' \notin E$:
  We first observe that $c' \neq a$ as $cc' \notin E$. If $ac' \notin E$, then $ba\Gamma bc'$. Thus, ba and bc' are in the same implication class. Thus, $A = C^{-1}$ which we ruled out. So $ac' \in E$ and $ac\Gamma ac'$, so $ac' \in B$.

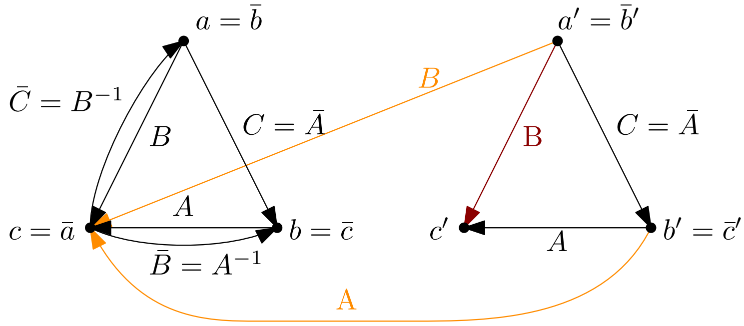


- Case $c = c', bb' \notin E$:
  We use a similar argument. We first observe that $b' \neq a$ as $bb' \notin E$. If $ab' \notin E$, then $b'c\Gamma ac$. Thus, $b'c$ and ac are in the same implication class. Thus, $A = B$ which we ruled out. So $ab' \in E$ and $ab\Gamma ab'$, so $ab' \in B$.

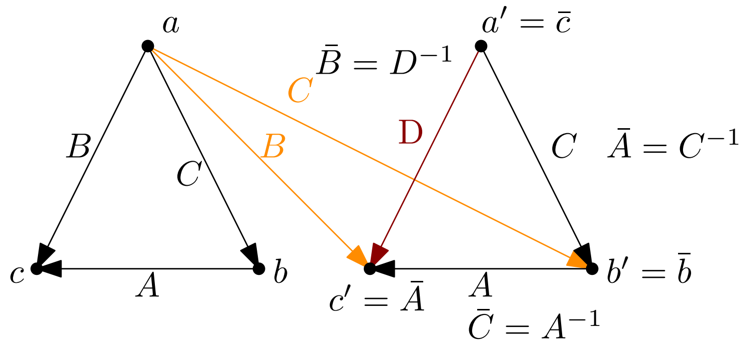


(ii) We apply (i) to $\bar{a}, \bar{b}, \bar{c}, \bar{A}, \bar{B}, \bar{C}$. We verify $C = \bar{A} \neq \bar{B} = A^{-1}$. But $C = \bar{A} \neq \bar{C}^{-1} = (B^{-1})^{-1} = B$ may be not true. We consider two cases.

- Case $B \neq C$:
  We apply part (i) and get $a'c \in B, b'c \in A$. By applying part (i) again to the triangle $a'b'c$ and alternative base $c'b'$ we get $a'c' \in B$.



- Case $B = C$:
  We use part (i) to obtain $ab' \in C, ac' \in B$. Again if $ac' \notin E$, then $b'a'\Gamma b'c'$ and $A = C^{-1}$. So $ac' \in E$.
  We still have to find the implication class of this edge. Now let $a'c' \in D \in \mathcal{I}(G)$. We assume $D \neq B$, or we are done. We apply part (i) to $\bar{a}, \bar{b}, \bar{c}$. We verify $B^{-1} = C^{-1} = \bar{A} \neq \bar{B} = D^{-1}$ and $C^{-1} = \bar{A} \neq \bar{C}^{-1} = A$. We use $ba \in \bar{A} = C^{-1}$ as the alternative base. This gives ac'. But then we have $B = D$.

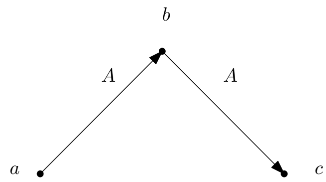


◀

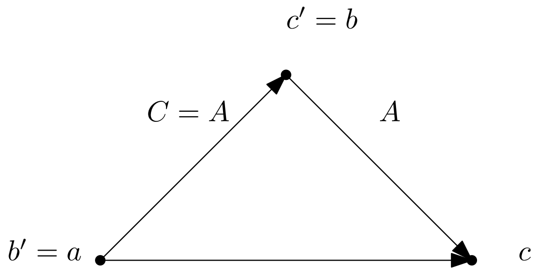
We continue by proving that implication classes are transitive in our cases.

▶ **Theorem 59** (Theorem 4.4.). *$A \in \mathcal{I}(G) \Rightarrow A = A^{-1}$ or $A \cap A^{-1} = \emptyset$ and A, A^{-1} transitive*

Proof. We know that $A = A^{-1}$ or $A \cap A^{-1} = \emptyset$ (Theorem 56). In the case $A \cap A^{-1} = \emptyset$ we show that A is transitive.

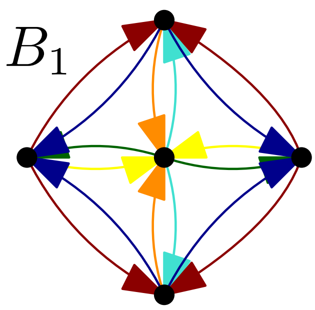


In this scenario we need to show $ac \in A$. If $ac \notin E$, then $ab \Gamma cb$ and due to $ab \in A$ and $cb \in A^{-1}$ we have $A = A^{-1}$. This contradicts our current case. So $ac \in E$. We consider an implication class $B \in \mathcal{I}(G)$ such that $ac \in B$ and show $B = A$. For this we assume $B \neq A$ for the sake of contradiction. We apply the triangle lemma part (i) with $b'c' = ab$ as the new base. We note that we can apply the lemma as $A \neq B$ and $A \neq C^{-1} = A^{-1}$. We thus get $ac' = ab \in B$ and thus $A = B$. This is a contradiction.



◀

We use this result to recognize comparability graphs. As this result shows that an implication class is transitive we can add an arbitrary implication class to the orientation. We then remove the full color class.



Consider the above graph. After removing B_1 one may choose orange and yellow classes but not green and orange. So some dependencies exist.

In the following we consider Algorithm 7 and prove its correctness.

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

$T \leftarrow \emptyset$;
$i \leftarrow 1$; $E_i \leftarrow E$;
while $E_i \neq \emptyset$ **do**
  choose $x_i y_i \in E_i$ arbitrarily;
  determine implication class B_i of E_i containing $x_i y_i$;
  if $B_i \cap B_i^{-1} \neq \emptyset$, **then**
    return *"G is no comparability graph"*;
  end if
  add B_i to T;
  $E_{i+1} \leftarrow E_i - \hat{B}_i$;
  $i \leftarrow i + 1$;
end while
return T;

Algorithm 7 : Recognition of comparabilty graphs

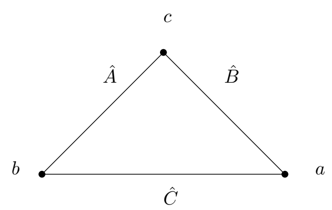
We formalize this notion of iteratively removing color classes.

▶ **Definition 60.** $[B_1, \ldots, B_k]$ *is a G-decomposition, if*

- $\hat{B}_1 + \cdots + \hat{B}_k = E$
- $B_i \in \mathcal{I}(\hat{B}_i + \cdots + \hat{B}_k)$ *for* $i \in [k]$

We note that Algorithm 7 computes a G-decomposition or stops with *not a comparability graph.* To prove the algorithms correctness we first investigate how implication classes change when removing color classes. Here, we introduce Theorem 4.6 that states that in this case either the color classes are independent and the order of removal could have been changes or two former classes were merged.

This theorem uses rainbow triangle which are structures similar to triangles, but care only about color classes.



Here, we require $\hat{A}, \hat{B}$ and $\hat{C}$ to be pairwise distinct.

▶ **Theorem 61** (Theorem 4.6). *For* $A \in \mathcal{I}(G), D \in \mathcal{I}(G - \hat{A})$ *we have*

(i) $D \in \mathcal{I}(G)$ *and* $A \in \mathcal{I}(G - \hat{D})$
(ii) $D = B + C, \hat{A}, \hat{B}, \hat{C}$ *in rainbow triangle*

Proof. We first note that all edges in Γ-relation before the removal of $\hat{A}$ are still in relation afterwards. But removing the color class may introduce additional relations as it introduces non-edges. So implication classes can merge. We consider $D \in \mathcal{I}(G - \hat{A})$ which is a disjoint union of some previous implication classes.

- case 1: $D = B + C + \dots, B, C \in \mathcal{I}(G)$:
  We show that in this case only two classes merge. In this case there must have been a rainbow triangle $\hat{A}\hat{B}\hat{C}$. If B also merges with X, then there must be a rainbow triangle $\hat{A}\hat{B}\hat{X}$. We then apply the triangle lemma part (ii) to get $\hat{X} = \hat{C}$. So $D = B + C$.
- case 2: $D \in \mathcal{I}(G)$
  By case 1 we know that every implication class of $\mathcal{I}(G - \hat{D})$ is a union of at most two implication classes of $\mathcal{I}(G)$. If A merges with X in $G - \hat{D}$, then there is a rainbow triangle $\hat{A}\hat{D}\hat{X}$. But then D merges with X or X^{-1} in $G - \hat{A}$. This is a contradiction. So $A \in \mathcal{I}(G - \hat{D})$.

◀

We are now ready to show our main theorem.

▶ **Theorem 62.** *The following statements are equivalent:*

(i) *G is a comparability graph*

(ii) $A \cap A^{-1} = \emptyset, \forall A \in \mathcal{I}(G)$

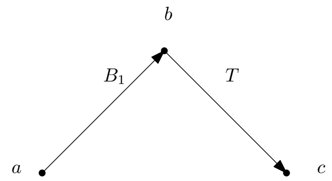
(iii) *Every G-decomposition* $[B_1, \dots, B_k]$ *has* $B_i \cap B_i^{-1} = \emptyset, \forall i \in [k]$

Note that every graph may have a G-decomposition but these may not fulfil the niceness-criterion.

Proof.
- (i)⇒(ii) is done by Theorem 56.
- (ii)⇒(iii)
  We consider any G-decomposition $[B_1, \dots, B_k]$ and use induction on k. For $k = 1$ we have $B_1 \in \mathcal{I}(G)$ so $B_1 \cap B_1^{-1} = \emptyset$ by (ii). For $k \geq 2$ we again have $B_1 \cap B_1^{-1} = \emptyset$ by (ii). We note that $[B_2, \dots, B_k]$ is a G-decomposition if $G - \hat{B}_1$. We need to verify (ii) for this graph, namely $A \cap A^{-1}, \forall A \in \mathcal{I}(G - \hat{B}_1)$. By Theorem 61 we have $D \in \mathcal{I}(G)$ and then $D \cap D^{-1} = \emptyset$ by (ii). Alternatively we have $D = B + C$ for $B, C \in \mathcal{I}(G)$. Then:

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

  Here, the first and last are empty due to (ii) and the other two are empty as $B \neq C^{-1}$ and $C \neq B^{-1}$. This is as there is a rainbow triangle $\hat{B}_1\hat{B}\hat{C}$ and implication classes are either the same or disjoint.
  So (ii) holds for $G - \hat{B}_1$ and by induction $B_i \cap B_i^{-1} = \emptyset, \forall i \geq 2$.
- (iii)⇒(i)
  We again use induction on k for $[B_1, \dots, B_k]$. For $k = 1$ we have $B_1 \cap B_1^{-1} = \emptyset$ and thus by Theorem 59 B_1 is a transitive orientation. The orientation part is due to the fact that $\hat{B}_1$ contains all edges. For $k \geq 2$ we consider the G-decomposition $[B_2, \dots, B_k]$ of $G - \hat{B}_1$. As this fulfils (iii) the graph $G - \hat{B}_1$ has a transitive orientation T by induction. We claim that $B_1 + T$ is a transitive orientation of G. The orientation part follows easily from the fact that B_1 orients all edges added to $G_{G - \hat{B}_1}$. We show transitivity. As transitivity can only break when edges of different parts are involved we consider the two possible cases.



If $ac \notin E$, then $ab\Gamma cb$. Then, $cb \in B_1$. This contradicts $bc \in T$. So $ac \in E$. But this edge may be oriented in the wrong direction. Then, ca is either in T or B_1. If $ca \in T$, then T is not transitive as ba is missing. If $ca \in B_1$, then B_1 is not transitive as cb is missing. So one must orient ac. The case where B_1 and T switch positions follows analogous. ◀

This proves our algorithm correct.

▶ **Corollary 63.** *Algo 7 determines correctly whether G is a comparability graph in $\mathcal{O}(\Delta(G) \cdot |E| + |V|)$.*

This is as the algorithm stops when is finds a not nice decomposition. We can analogously show that the above T is in fact the $B_2 \cup \cdots \cup B_k$ computed by the algorithm. For the runtime the critical line is line 5. This can be done by exploring the neighbours of the starting edge. This contributes the factor of the maximal degree $\Delta(G)$.

We are now ready to state an algorithms computing our parameters.

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

```
compute transitive orientation F of G;
compute tological ordering σ of (V, F);
for i ← 1 to n do
    v ← σ(i);
    h(v) ← 1 + max{h(w) | wv ∈ F};
    χ ← max{χ, h(v)};
    w ← argmax{h(w), h(v)};
end for
for i ← χ to 1 do
    C ← C + {w};
    w ← argmax{h(v) | vw ∈ F};
end for
return h and C;
```

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

▶ **Theorem 64.** *Algorithm 8 computes correctly $\chi(G), \omega(G)$ for a comparability graph G in $\mathcal{O}(|V| + |E|)$ (when given a transitive ordering).*

Proof. We show the partial aspects.

- h is a coloring:
  $\forall uv \in E(G)$ to show: $h(u) \neq h(v)$. We assume w.l.o.g. $uv \in F$. So since σ is a topological ordering we have $\sigma(u) < \sigma(v)$. So $h(u)$ is set already when $v = \sigma(i)$. Thus, $h(v) = 1 + \max\{h(w) | wv \in F\} \geq 1 + h(u)$. As this includes u we have the desired outcome. We thus know $\chi = \max_{v \in V} h(v) \geq \chi(G)$.

- C is a clique:
  Let $C = \{w_\chi, w_{\chi-1}, \ldots, 1\}$. Then, $h(w_{\chi-i}) = \chi - i$ and $h(w_{\chi-i}) = 1+\max\{h(v)|vw_{\chi-1} \in F\}$. So $h(w_{\chi-i-1}) = \chi - i - 1$ and $w_{\chi-i-1}w_{\chi-i} \in F$. So C is a directed path in F. By transitivity C is a clique. So we have $\chi = |C| \leq \omega(G)$.
- The coloring and clique are optimal:
  We combine these results to get: $\chi(G) \leq \chi = |C| \leq \omega(G) \leq \chi(G)$.
- The runtime is linear except for line 1.

◀

We now introduce Algo which computes $\alpha(G)$ and $\kappa(G)$ for comparability graphs. We first consider the special case of bipartite graphs. Remember that these are comparability graphs as we can orient all edges from the first to the second set.

We introduce some terminology.

▶ **Definition 65.** *A set $M \subseteq E_G$ is a* matching*, if $\forall v \in V_G$ there is at most one $e \in M$ with $v \in e$.*

We note that $\kappa(G) \leq |V| - \max\{|M| : M \text{ Matching}\}$ as each edge of the matching improves the trivial clique cover of isolated vertices by one.

▶ **Definition 66.** *A set $S \subseteq V_G$ is a* vertex cover*, if $\forall e \in E_G$ there is at least one $v \in S$ with $v \in e$.*

We note that S is a vertex cover if and only if $V - S$ is an independent set. So we have $\alpha(G) \leq |V| - \min\{|S| : S \text{ vertex cover}\}$. These two hold for all graphs as well as $\alpha(G) \leq \kappa(G)$.

▶ **Theorem 67** (König)**.** *For a bipartite graph G we have $\min\{|S| : S \text{ vertex cover}\} = \max\{|M| : M \text{ Matching}\}$.*

Proof. We need to show that the two inequalities are in truth equalities. As G is perfect we have $\alpha(G) = \kappa(G)$ and it is clear that we can use a maximal matching to find a minimal clique cover by using the matching edges and the remaining isolated vertices. ◀

So on bipartite graphs we can find the desired properties by computing a maximal matching.

For other graphs we construct a bipartite auxiliary graph $B = (V', V'', E)$. This graph uses two copies of V as vertices. So $V' = \{v'|v \in V\}, V'' = \{v''|v \in V\}$ and $vw \in F \Leftrightarrow v'w'' \in E$. Here, we note a correspondence between clique covers of G and matchings of B. We use the following rule: v, w consecutive in the clique cover $\Leftrightarrow v'w'' \in M$. Here, two vertices are consecutive, if the are neighbours on the oriented path in a clique of the cover.

clique cover $V_1 + \cdots + V_k$ of G
$\Leftrightarrow k$ cliques partition V_G
$\Leftrightarrow k$ distinct paths in F partition G
$\Leftrightarrow 2 \cdot k$ starts and ends of paths
$\Leftrightarrow 2 \cdot k$ vertices of B are unmatched
$\Leftrightarrow 2 \cdot |V_G| - 2 \cdot k = 2 \cdot |M|$
$\Leftrightarrow |V_G| - k = |M|$

So $\kappa(G) = |V| - \max\{|M| : M \text{ Matching in } B\} = |V| - \min\{|S| : S \text{ vertex cover in } B\}$ using the Königs theorem.

To show optimality consider a vertex cover S of B. We note that for all $v \in S$ the set $S - v$ is not a vertex cover due to the minimality. We then use the following observation:

▶ **Observation 68.** $|S \cap \{v', v''\}| \leq 1, \forall v \in V_G$.

Proof. We assume the contrary. Since $S - v'$ is no vertex cover: $\exists w \in V_G, w' \notin S, vw \in F$. Since $S - v''$ is no vertex cover: $\exists u \in V_G, u'' \notin S, uv \in F$. By transitivity there is $uw \in F$ and thus there is an uncovered edge $u'w'' \in E_B$. This is a contradiction. ◀

Hence, $Y = \{v \in V_G | S \cap \{v', v''\} = \emptyset\}$, the set of all vertices where neither copy is covered, has exactly $|V_G| - |S| = |V_G| - |M| = \kappa(G)$ elements.

▶ **Observation 69.** *Y is an independent set in G.*

Proof. We assume $vw \in E_G$ and have w.l.o.g. $vw \in F$. Then, $v'w'' \in E_B$, but $S \cap \{v', w''\} = \emptyset$. So there is an uncovered edge and thus a contradiction. ◀

Hence, $\alpha(G) \geq |Y| = \kappa(G) \geq \alpha(G)$. So $|Y| = \alpha(G)$.

So we obtain the following algorithm 9.

1. compute a transitive orientation F
2. compute the bipartite graph B
3. compute a maximal matching M in B
4. compute a minimal vertex cover S in B from M
5. derive clique cover of $|V| - |M|$ cliques
6. derive independent set on $|V| - |S|$ vertices

Here, the first step takes $\mathcal{O}((|V|+|E|)^2)$, the third takes $\mathcal{O}(|E|^{1,5})$ (with modern algorithms nearly linear) and all others take $\mathcal{O}(|V| + |E|)$.

5 Graph classes derived from chordal and comparability graphs

We first characterize split graphs.

▶ **Theorem 70** (Theorem 5.3.). *For a graph $G = (V, E)$ the following are equivalent.*

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

(ii) $V = K + S$ with K being a clique and S being an independent set

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

Proof. We show the following three implications.

- (ii)⇒(i):
  We have $V = K + S$ such that K is a clique and S is independent. Then, let C be a cycle of length at least four. If $V(C) \cap S = \emptyset$, then C has a chord as K is a clique. If $C = [v_1, v_2, v_3, v_4, \dots]$ with $v_2 \in S$, then $v_1v_3 \in K$ as S is an independent set. So $v_1v_3 \in E$ and C has a chord. Thus, G is chordal. Analogously, $\overline{G}$ is also chordal as for $\overline{G}$ the set K is independent and S forms a clique.
- (i)⇒(iii):
  This implication has been shown before.
- (iii)⇒(ii):
  We find a split into K and S. For this we choose K as the maximum clique such that G_S for $S = V - K$ has the fewest edges. Assume that G_S has an edge xy. Then we find an induced C_4 or C_5 in G or an induced C_4 in $\overline{G}$. In the later case this is equivalent to finding an induced $2K_2$ in G. Since K is maximum there exists $u, v \in K$ such that

$ux \notin E$ and $vy \notin E$. If $u = v$ for all choices, then $K - u + \{x, y\}$ is a larger clique. This contradicts the maximality. So $u \neq v$. If we have $vx, uy \in E$, then a C_4 is induced. If we have $vx, uy \notin E$, then a $2K_2$ is induced. So we may assume w.l.o.g. that $vx \in E$ and $uy \notin E$. We then find split K' and S'. We now consider $K' = K - v + y$ and show that K' is a clique. So take $w \in K - u, v$. Assume that $wy \notin E$. If $wx \notin E$, then v, w, x and y form a $2K_2$. If $wx \in E$, then u, w, x and y form a C_4. So all such w are connected to y and thus K' forms a clique of the same size. We now show that $G_{V-K'}$ has fewer edges than G_{V-K}. To show this we prove $|\mathrm{Adj}(y) \cap S| > |\mathrm{Adj}(v) \cap S|$. We assume $t \in S, tv \in E, ty \notin E$. If $tx \notin E$, then xy, v and t form a $2K_2$. So $tx \in E$. If $tu \notin E$, then u, v, x and y form an induced C_5. If $tu \in E$, then u, t, x and y form an induced C_4. So no such t exists. Thus, $|\mathrm{Adj}(y) \cap S| > |\mathrm{Adj}(v) \cap S|$ and $G_{V-K'}$ has fewer edges. Then, this is a contradiction of the choice of G_{V-K}.

◀

Next, we introduce *permutation graphs*.

▶ **Theorem 71.** *For every undirected graph $G = (V, E)$ the following are equivalent:*

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

(ii) *There exists a vertex ordering σ of G without*  

(iii) *There exists an embedding $V \to \mathbb{R}^2$ such that ($uv \in E$) if and only if ($u_x < v_x \Leftrightarrow u_y < v_y$).*

Proof. We show the three implications.

- (i)⇒(ii):
  Since they are comparability graphs there exist transitive orientations (V, F_1) of G and (V, F_2) of $\overline{G}$. Then, $F = F_1 + F_2$ is an orientation of the complete graph on V. We claim that if F_1 and F_2 are transitive, then F is also transitive. The orientation of the complete graph is transitive if and only if F is acyclic. So if F is not transitive it contains directed cyclic triangles where either all edges are in one orientation or where one edge is in a different orientation from the rest. In either case the F_i with two edges is not transitive as well. So the claim is true. So let σ be a topological ordering of F. This exists due to the transitivity of F. Then, the first pattern contradicts the transitivity of F_1 and the second the transitivity of F_2.
- (ii)⇒(iii):
  We are given a σ without those patterns and can obtain a transitive orientation F_1 of G and F_2 of $\overline{G}$ by orienting left-to-right. We then take $\sigma_x = \sigma$ as the order of x-coordinates of points for each vertex in V. Since $F_1 + F_2^{-1}$ is also a transitive orientation of a complete graph, we can use a second ordering σ_y that is the topological ordering of $F_1 + F_2^{-1}$. So consider two vertices u, v. If $uv \in E$, we have w.l.o.g. $uv \in F_1$ and thus $u_x < v_x$ and $u_y < v_y$. If $uv \notin E$, we have w.l.o.g. $uv \in F_2$ and thus $u_x < v_x$ but $vu \in F_2^{-1}$ and $u_y > v_y$.
- (iii)⇒(i):
  Given an embedding of V in the plane we orientate $uv \in E$ from u to v if and only if u is to the bottom-left of v. This is transitive and thus G is a comparability graph. Analogously, we use the bottom-right for $\overline{G}$.

◀

We can also use an alternative definition for *permutation graphs*. Given an ordering π of $[n]$, we define $G = G_\pi$ as $V(G) = [n]$ and $ij \in E(G) \Leftrightarrow (i - j)(\pi(i) - \pi(j)) < 0$. That is G has an edge if π inverts the two vertices.

We can recognize permutation graphs in linear time and can compute χ, ω, α and κ due to the connection to comparability graphs. Furthermore χ and ω can be done in $\mathcal{O}(|V| + |E|)$.

We now consider a different problem. We are given intervals $I_1, \ldots, I_n$ with $I_i = (x_i, y_i)$ sorted such that $x_1 \leq x_2 \leq \cdots \leq x_n$. Our goal is to find the number of minimal translations such that these intervals do not intersect. So we want (i) $x'_1 \leq \cdots \leq x'_n$ and (ii) $y'_i < x'_{i+1} \forall i \in [n-1]$. We construct a conflict graph G with $V(G) = \{I_1, \ldots, I_n\}$ and $I_i I_j \in E(G) \Leftrightarrow x_j - y_j < \sum_{i<k<j} (y_k - x_k)$. We can show that G is a permutation graph and that the maximal set of intervals that are not moved is a maximum independent set.

We are now ready to return to interval graphs and characterize them through chordal and comparability graphs.

▶ **Theorem 72.** *For every $G = (V, E)$ the following are equivalent.*

(i) *G is an interval graph*

(ii) *there exists a vertex ordering σ without* 

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

(iv) *G has no induced C_4 and $\overline{G}$ is a comparability graph*

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

Proof. We show the implications.

- (i)⇒(ii):
  We look at the interval representation of the graph. We may assume w.l.o.g. that all endpoints of the intervals are distinct. We then define σ as the left to right ordering of these endpoints. So let $u <_\sigma v <_\sigma w$ with $uw \in E$, then the interval of v ends between the endpoints of the other two intervals. Since $uw \in E$ the interval of w intersects the one of u and thus the one of v.
- (ii)⇒(iii):
  The ordering σ has none of the triplets characterizing chordal graphs as they are forbidden by the above triplet. Similarly, complement of the forbidden triplet of comparability graphs is part of the above triplet.
- (iii)⇒(iv):
  This is trivial.
- (iv)⇒(v):
  We know that C_4 is no induced subgraph of G and that $\overline{G}$ is a comparability graph. So $2K_2$ is no induced subgraph of $\overline{G}$. Then, let F be a transitive orientation of $\overline{G}$ and let A, B be inclusion-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 $B < A$. It can be shown that if $A < B$, then NOT $B < A$ by case distinction. So $<$ is well-defined. We now need to show that $<$ is acyclic.
  So let $A < B < C$ and show $A < C$. Then, let $a_1 b_1 \in F, a_1 \in A, b_1 \in B$ and let $a_2 b_2 \in F, a_2 \in B, b_2 \in C$. We know $b_2 \notin A$ as otherwise there would be a non-edge from B to A. If $b_1 = a_2$, we also have $a_1 b_2 \in F$ by transitivity and thus $A < C$. So assume $b_1 \neq a_2$. Then, $b_1 a_2 \in E$ as B is a clique and $a_1 a_2 \in E, b_1 b_2 \in E$ as otherwise transitivity must be violated. Since C_4 is no induced subgraph of G, we know that $a_1 b_2 \notin E$. Due to the transitivity we get $a_1 b_2 \in F$ and thus $A < C$. So there is an total order $A_1 < \cdots < A_x$ on maximal cliques.
  Let $v \in A_i \cap A_k$ with $i < j < k$. We have to show that $v \in A_j$. We assume that $v \notin A_j$ and show a contradiction. Then, there is a vertex $w \in A_j$ with $vw \notin E$ as A_j is

inclusion-maximal and does not contain v. If $vw \in F$, then $A_k < A_j$. This contradicts $j < k$. If $wv \in F$, then $A_j < A_i$. This contradicts $i < j$. So $v \in A_j$.

(v)⇒(i):
We are given an ordering $A_i, \ldots, A_x$ on inclusion-maximal cliques. We note that $\{i|v \in A_i\}$ is an interval. So let I_v be the smallest interval such that $\{i|v \in A_i\} \subseteq I_v$. So $vw \in E \Leftrightarrow \exists i : vw \in A_i \Leftrightarrow I_v \cap I_w = \emptyset$. Here, the first equality is due to the fact that each edge forms a clique and thus two neighbours must be in atleast one inclusion-maximal clique together. So G is an interval graph.

◀

We end with a short overview on these different graph classes.

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.

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	???