Algorithmic Graph Theory

Solution Sheet 6

Laura Merker and Samuel Schneider, July 30, 2025

Exercises

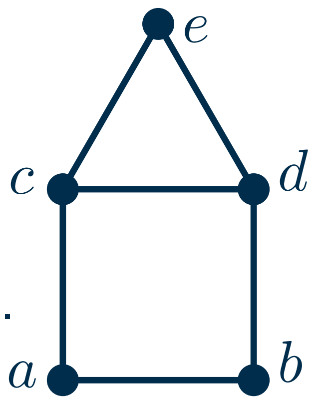
Let G be a comparability graph and $[B_1, \ldots, B_k]$ a G-decomposition. A tupel $(e_1, \ldots, e_k)$ of edges is called a ***decomposition scheme*** of G if there is a G-decomposition $[B_1, \ldots, B_k]$ such that $e_i \in B_i$ for all $i \in [k]$.

(1) Prove that every permutation of a decomposition scheme is a decomposition scheme as well.

(2) Let G be a comparability graph. Prove that every G-decomposition has the same length.

(3) Give an example for each combination of chordal, co-chordal, comparability graph and co-comparability graph.

(4) Find a matching representation for the graph on the bottom. Is there a matching representation such that the vertices in the top row are ordered a, b, c, d, e?



(5) Which trees are permutation graphs?

(6) Prove that the rainbow number and the queue number are equal for every graph.

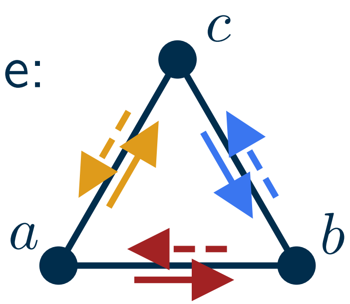
Exercise 1

Let $(e_1, \ldots, e_k)$ be a decomposition scheme and $[B_1, \ldots, B_k]$ corresponding G-decomposition.

For $k = 1$ the statement holds trivially. Thus, let $k \geq 2$ and $i < k$.

Notation:
$E_i = \hat{B}_i + \ldots + \hat{B}_k$
C_i: implication class in E_i s.t. $e_{i+1} \in C_i$
C_{i+1}: implication class in $E_i - \hat{C}_i$ s.t. $e_i \in C_{i+1}$

Example:



(ab, ac) is a scheme with $B_1 = \{ab\}, B_2 = \{ac, bc\}$.

(ac, ab) is a scheme with $C_1 = \{ac\}, C_2 = \{ab, cb\}$.

Exercise 1

Let $(e_1, \ldots, e_k)$ be a decomposition scheme and $[B_1, \ldots, B_k]$ corresponding G-decomposition.

For $k = 1$ the statement holds trivially. Thus, let $k \geq 2$ and $i < k$.

Notation:
$E_i = \hat{B}_i + \ldots + \hat{B}_k$
C_i: implication class in E_i s. t. $e_{i+1} \in C_i$
C_{i+1}: implication class in $E_i - \hat{C}_i$ s. t. $e_i \in C_{i+1}$

Thm 4.6 $A \in \mathcal{I}(G), D \in \mathcal{I}(E - \hat{A})$

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

or **(ii)** $D = B + C$, $\hat{A}, \hat{B}, \hat{C}$ in rainbow triangle

Goal: $[B_1, \ldots B_{i-1}, C_i, C_{i+1}, B_{i+2}, \ldots B_k]$ is G-decomposition.

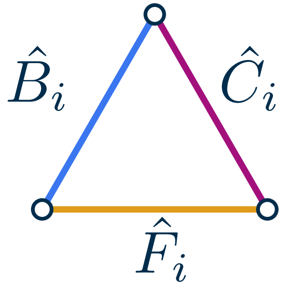
By Theorem 4.6 and $A = B_i$ and $D = B_{i+1}$ we have:

(i) $B_{i+1} \in \mathcal{I}(E_i)$ and $B_i \in \mathcal{I}(E_i - \hat{B}_{i+1})$ or

- Then, $B_{i+1} = C_i$ and $B_i = C_{i+1}$ and it holds that $\hat{B}_i + \hat{B}_{i+1} = \hat{C}_i + \hat{C}_{i+1}$.

(ii) $B_{i+1} = C_i + F$, $\hat{B}_i, \hat{C}_i, \hat{F}$ in rainbow triangle

- Then, $C_{i+1} = B_i + F$ or $C_{i+1} = B_i + F^{-1}$.
- Thus, $\hat{C}_{i+1} = \hat{B}_i + \hat{F}$ and therefore $\hat{B}_i + \hat{B}_{i+1} = \hat{C}_i + \hat{C}_{i+1}$.



Exercise 1

Let $(e_1, \ldots, e_k)$ be a decomposition scheme and $[B_1, \ldots, B_k]$ corresponding G-decomposition.

For $k = 1$ the statement holds trivially. Thus, let $k \geq 2$ and $i < k$.

Notation:
$E_i = \hat{B}_i + \ldots + \hat{B}_k$
C_i: implication class in E_i s. t. $e_{i+1} \in C_i$
C_{i+1}: implication class in $E_i - \hat{C}_i$ s. t. $e_i \in C_{i+1}$

Thm 4.6 $A \in \mathcal{I}(G), D \in \mathcal{I}(E - \hat{A})$

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

or **(ii)** $D = B + C$, $\hat{A}, \hat{B}, \hat{C}$ in rainbow triangle

Goal: $[B_1, \ldots B_{i-1}, C_i, C_{i+1}, B_{i+2}, \ldots B_k]$ is G-decomposition.

By Theorem 4.6 and $A = B_i$ and $D = B_{i+1}$ we have:

(i) $B_{i+1} \in \mathcal{I}(E_i)$ and $B_i \in \mathcal{I}(E_i - \hat{B}_{i+1})$ or

(ii) $B_{i+1} = C_i + F$ with $F \in \mathcal{I}(E_i)$ and $\hat{B}_i, \hat{C}_i, \hat{F}$ rainbow triangle

- In both cases we have $\hat{B}_i + \hat{B}_{i+1} = \hat{C}_i + \hat{C}_{i+1}$.
- Thus, $E = \hat{B}_1 + \ldots \hat{B}_{i-1} + \hat{C}_i + \hat{C}_{i+1} + \hat{B}_{i+2} + \ldots \hat{B}_k$.
- Therefore, $[B_1, \ldots, B_{i-1}, C_i, C_{i+1}, B_{i+2}, \ldots, B_k]$ is G-decomposition with scheme $(e_1, \ldots, e_{i-1}, e_{i+1}, e_i, e_{i+2}, \ldots, e_k)$.

We can obtain all permutations by repeating this.

Exercise 2

Let $(e_1, \dots, e_k)$, $(f_1, \dots, f_l)$ be schemes, $[B_1, \dots, B_l]$ G-decomposition w. r. t. (f_i).

Lemma:
There is $j \in [l]$ s. t. $(f_1, \dots, f_{j-1}, e_1, f_{j+1}, \dots f_l)$ is scheme of G.

- Since $E = \hat{B}_1 + \dots + \hat{B}_l$ there is $j \in [l]$ s. t. $e_1 \in \hat{B}_j$.
- If $e_1 \in B_j$ / $e_1 \in B_j^{-1}$ then $[B_1, \dots, B_{j-1}, B_j, B_{j+1}, \dots B_l]$ / $[B_1, \dots, B_{j-1}, B_j^{-1}, B_{j+1}, \dots B_l]$ is G-decomposition w. r. t. $(f_1, \dots, f_{j-1}, e_1, f_{j+1}, \dots f_l)$.

Exercise 2

Let $(e_1, \dots, e_k)$, $(f_1, \dots, f_l)$ be schemes, $[B_1, \dots, B_l]$ G-decomposition w. r. t. (f_i).

Induction on the number of color classes of G.

If $E = \hat{A}$ for an implication class A then $k = l = 1$ and every edge can be chosen.

Let $k, l \geq 2$.

Idea: Add e_1 to (f_i) and delete color class of e_1.

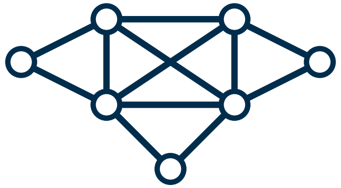
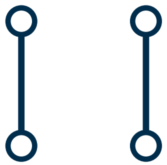
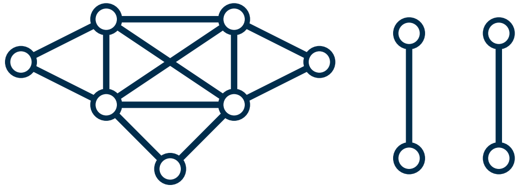
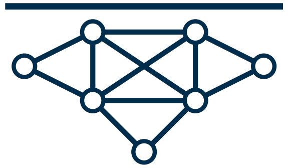
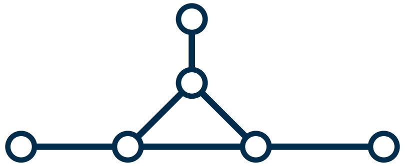
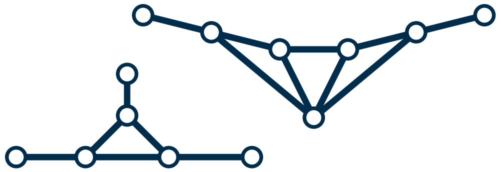
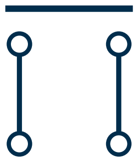
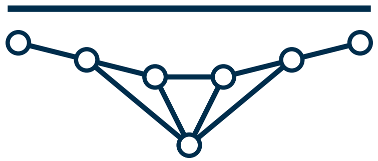
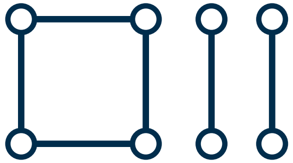
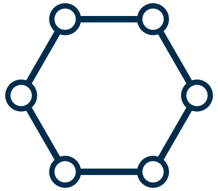
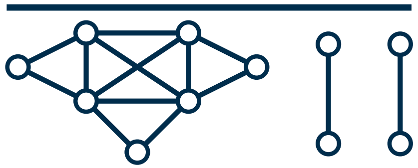
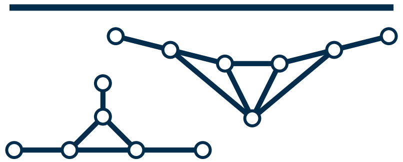
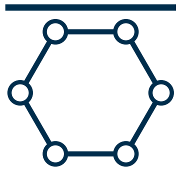
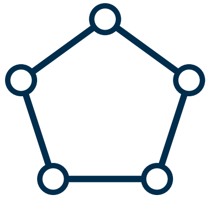
Lemma from before: $\exists j \in [l]$ s. t. $(f_1, \dots, f_{j-1}, e_1, f_{j+1}, \dots f_l)$ is scheme.

By exercise 1: $(e_1, f_1, \dots, f_{j-1}, f_{j+1}, \dots, f_l)$ is scheme.

- Now we have **two** schemes of G with e_1 as the first edge.
- Deleting color class of e_1 in G results in graph $G - \hat{B}$ with schemes $(e_2, \dots, e_k)$ and $(f_1, \dots, f_{j-1}, f_{j+1}, \dots, f_l)$.
- $G - \hat{B}$ has less color classes than G. By induction we have $k - 1 = l - 1$.

Exercise 3

obtain bottom-left of table by taking complement of graphs on top-right

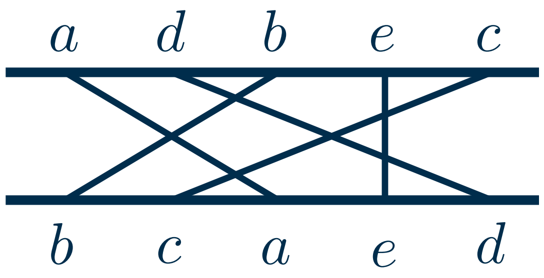
G \ $\overline{G}$	chordal & comp.	chordal & ¬comp.	¬chordal & comp.	¬chordal & ¬comp.
chordal & comp.				
chordal & ¬comp.				
¬chordal & comp.				
¬chordal & ¬comp.				

Exercise 4

Matching representation:

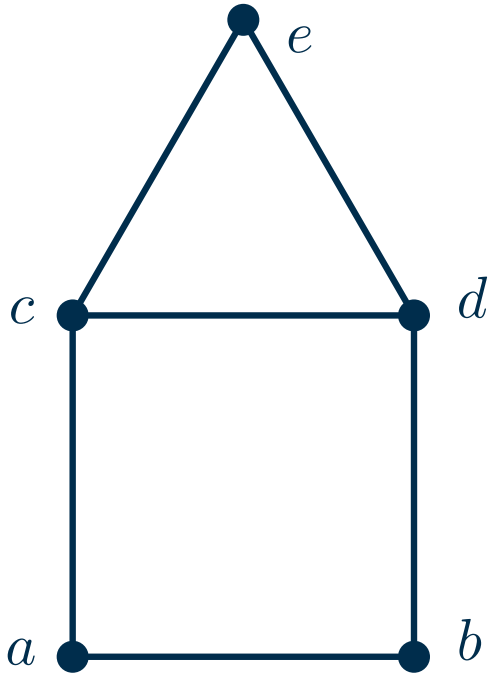
Observation: aeb not possible in top row

- a and b have to be inverted.
- Then, a or b are inverted with e.



Verification:

- d is with everything but a inverted.
- c is with everything but b inverted.
- a and b are with each other but not e inverted.



Possible with top row $abcde$?

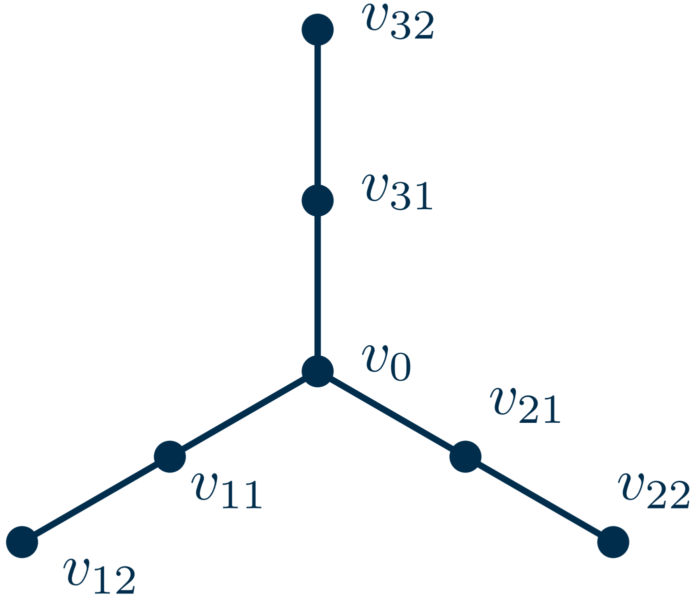
- Incident (non-)edges to e force:

$$\{a,b\} \quad e \quad \{c,d\}$$

- b and d have to be inverted ↯

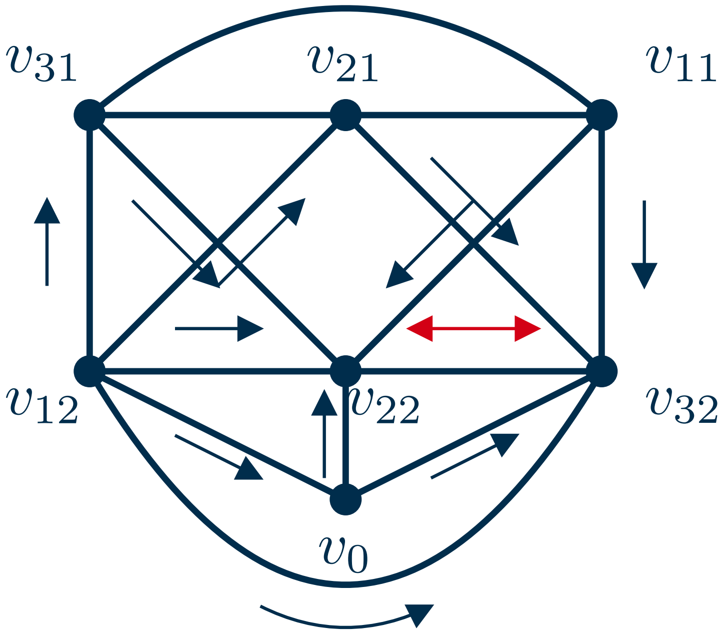
Exercise 5

We show that a spider is not a permutation graph by showing that its complement is not transitively orientable.



spider: at least three legs, each of length at least 2

$v_{32}v_{22} \;\Gamma\; v_{31}v_{22}$
$v_{31}v_{22} \;\Gamma\; v_0v_{22}$
$v_0v_{22} \;\Gamma\; v_{11}v_{22}$
$v_{11}v_{22} \;\Gamma\; v_{12}v_{22}$
$v_{12}v_{22} \;\Gamma\; v_{12}v_{21}$
$v_{12}v_{21} \;\Gamma\; v_{12}v_0$
$v_{12}v_0 \;\Gamma\; v_{12}v_{31}$
$v_{12}v_{31} \;\Gamma\; v_{12}v_{32}$
$v_{12}v_{32} \;\Gamma\; v_{11}v_{32}$
$v_{11}v_{32} \;\Gamma\; v_0v_{32}$
$v_0v_{32} \;\Gamma\; v_{21}v_{32}$
$v_{21}v_{32} \;\Gamma\; v_{22}v_{32}$



$\Rightarrow v_{32}v_{22}\Gamma^* v_{22}v_{32}$

$\Rightarrow$ spiders are not permutation graphs

Exercise 5

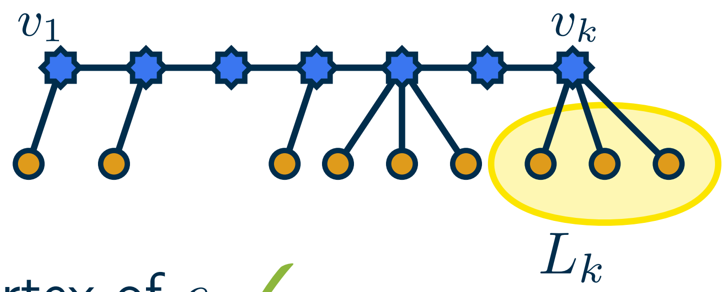
Fun Fact: Spider-free trees are called caterpillars

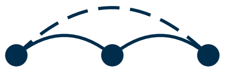
Claim: A tree is a permutation graph if and only if it is spider-free.

"$\Rightarrow$" Permutation graphs are hereditary, so they do not contain spiders as induced subgraphs

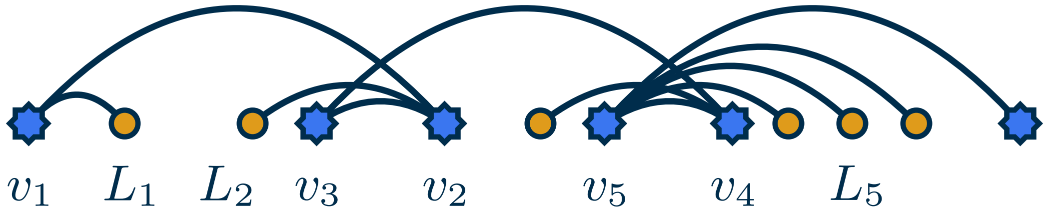
"$\Leftarrow$" Let T be a spider-free tree.

$V' = \{v \in V : \deg(v) \geq 2\}$ induces a path $v_1, \ldots, v_k$ in T.



Idea: Find a vertex order without  and 

- Every vertex below an edge e has to be connected to an end vertex of e. ✓
- Every vertex has edges in only one direction. (stronger statement) ✓



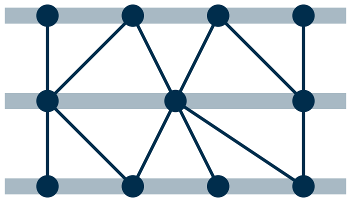
For $i \in [k]$ let L_i be the leaves adajcent to v_i.

Queue Layouts

- For a fixed linear order $\prec$ of a graph, we say that two edges vw, xy with $v \prec w$ and $x \prec y$ ***nest*** if $v \prec x \prec y \prec w$ or $x \prec v \prec w \prec y$.
- A ***rainbow*** (w.r.t. a linear vertex order) is a set of edges that are pairwise nesting.
- The ***rainbow number*** of a graph is the smallest k such that there is a linear vertex order whose largest rainbow has size at most k.
- The ***queue number*** of a graph is the smallest k such that there is a linear vertex order and a partition of the edges into at most k sets such that no two edges in the same part nest.

Prove that the rainbow number and the queue number are equal for every graph.

Warm-up: Find families of graphs with small (constant), respectively large (unbounded), queue number / rainbow number.

level planar graphs  complete graphs

Queue Layouts

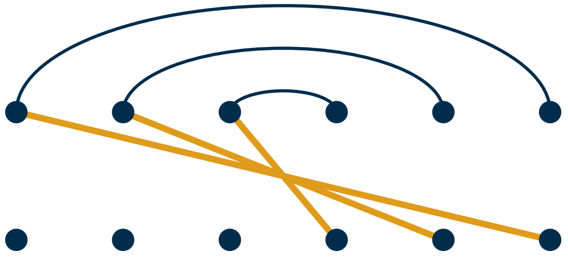
$\mathrm{qn}_{\prec}(G)$ = min # of colors such that there is an edge coloring without monochromatic 

We prove: $\mathrm{qn}_{\prec}(G) = \mathrm{rn}_{\prec}(G)$ for all G and $\prec$

$\mathrm{rn}_{\prec}(G)$ = min size of a rainbow w.r.t. $\prec$

transform ordered graph to matching

Observation: two edges nest w.r.t $\prec$ $\iff$ they cross in the matching representation Γ



queue layout w.r.t. $\prec$

matching representation Γ:
startpoints at the top,
endpoints at the bottom

Conflict graph H with $V(H) = E(G)$,
$E(H) = \{e_1 e_2 \mid e_1, e_2 \text{ cross in } \Gamma\}$

- H is a permutation graph
- $\omega(H) = \chi(H)$

colors in Γ transfer to queue layout

We conclude: $\mathrm{rn}_{\prec}(G)$ = max # of pw crossing edges in $\Gamma = \omega(H) = \chi(H) = \mathrm{qn}_{\prec}(G)$