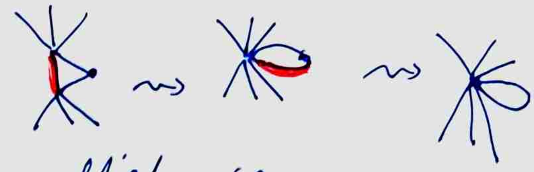
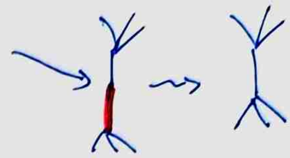
Graph relations

• edge contractions — multigraph contractions 

simple contractions ... no multiedges / loops

topological contractions – undo edge subdivisions

↳ one of the vertices has degree $= 2$ → 

→ we will generarily not consider multigraph cont.

Def: We say that a graph H is a

→ if H can be obtained from G via a series of operations:

1. subgraph of G ... $H \subseteq G$ ... edge & vertex deletions
2. induced subgraph ... $H \leq G$ ... vertex deletions
3. minor of G ... $H \leq_m G$ ... edge & vertex deletions and edge contractions
4. induced minor ... $H \leq_i G$ ... vertex deletions and edge contractions
5. topological minor ... $H \leq_t G$ ... edge & vertex deletions and topological contractions

All of these are partial orders on the class of all (finite) graphs

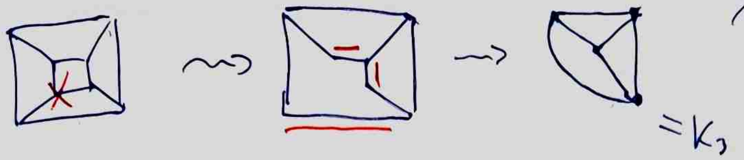
- reflexive ✓
- antisymmetric ... if $H \leq G$ & $H \neq G$, then $G \not\leq H$ ∵ H would need to grow
- transitive ... concatenate the series of operations

$H \leq G \Rightarrow H \subseteq G \Rightarrow H \leq_t G \Rightarrow H \leq_m G$

$\Downarrow H \leq_i G \Rightarrow H \leq_m G$

Examples: The reverses do not hold

- $\leq_m \not\Rightarrow \leq_t$: 
- $\leq_m \not\Rightarrow \leq_i$:  C_7 ... not ind. minor ∵ vertex del and contr. both remove 1 vertex ⇒ we can do at most 1 operation, and that doesn't make C_7
- $\leq_t \not\Rightarrow \leq_i$, $\subseteq \not\Rightarrow \leq_i$:  ... $\leq_i$ cannot delete edges
- $\leq_i \not\Rightarrow \leq_t$:  works as well

Ex:  $= K_3$

minor & topo-minor & ind. minor ∵ only vertex deletion and topological contraction

Lemma: We can first delete, then contract:

- $H \le_m G \iff H$ can be obtained from $G' \subseteq G$ via a series of contractions
- $H \le_i G \iff H$ can be obtained from $G' \le G$ via a series of contractions
- $H \le_t G \iff H$ can be obtained from $G' \subseteq G$ via a series of topo contractions
  $\iff$ a subdivision of H is a subgraph of G

$\overset{v}{\times}$ = del vertex
$\overset{e}{\times}$ = del edge
$\overset{c}{:}$ = contract

Proof: Given a sequence of operations $\overset{v}{\times}\ \overset{v}{\times}\ \overset{e}{\times}\ \overset{e}{\times}\ \overset{v}{\times}\ \overset{c}{:}\ \overset{c}{:}\ \overset{e}{\times}\ \overset{v}{\times}\ \overset{e}{\times}\ \overset{c}{:}\ \overset{v}{\times}\ \overset{c}{:}\ \overset{c}{:}\ \overset{e}{\times}$

we can move all edge deletions to the beginning of the sequence

→ now we only need to handle $\overset{v}{\times}$ after the first $\overset{c}{:}$

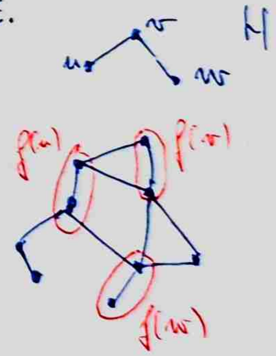
→ process from left to right. If v wasn't created by contraction (it is an original vertex), then we can simply delete v sooner and nothing will change

→ if v was created by contracting an edge, $u_1 - u_2$, delete u_1 and u_2, and apply recursion: if u_i was created by contraction, delete those vertices ...

⇒ $\sigma_1, \sigma_2, \sigma_3, c_1, c_2, c_3, \overset{v}{\times}$ ... suppose v was created by contracting $u_1 \underset{c_1}{-} u_2 \underset{c_3}{-} u_3$, then we get: $\sigma_1, \sigma_2, \sigma_3, \overset{u_1}{\times}\ \overset{u_2}{\times}\ \overset{u_3}{\times}, c_2$ ... we remove c_1, c_3 and get 3 deletions instead ■

Lemma: $H \le_m G$ resp. $H \le_i G$ (*) $\iff \exists$ mapping $f: V_H \to P(V_G)$ s.t.

1. $\forall u \in V_H$: $f(u)$ induces a nonempty connected subgraph of G
2. $\forall u \neq v \in V_H$: $f(u) \cap f(v) = \emptyset$ ... disjoint subgraphs
3. $\forall u, v \in V_H$: $uv \in E_H \implies \exists x \in f(u)\ \exists y \in f(v)$ s.t. $xy \in E_G$
   $\iff$ (*)



Note: The "blobs" $f(u)$ are called branch sets

Proof: $\Leftarrow$ remove unused vertices and edges and contract blobs

↪ for ind. minor, we do not need to remove any edges between $f(u), f(v)$ ∵ $\iff$

$\Rightarrow$: define v by "decontractions" ... we have H and want to create G

- $f(u) :=$ set of vertices from G that was merges into u by contractions

① ✓, ② ✓, ③ ✓ ∵ we start from H and there it was satisfied ■

Remark: This is how we can define minors for infinite graphs

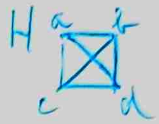
Note: If we want a similar characterization of topo minors, then we strengthen ① to require that the branch set with its adjacent vertices has a subdivision of a star as a subgraph such that the ends of the rays of the star are outside →

Application: If $H \le_m G$, then we can assume that H was constructed from a subgraph $G' \subseteq G$ via contractions

→ we can divide G' into branch sets $f(u)$ (✱) and keep at most 1 edge between $f(u)$ and $f(v)$

now: $uv \in E_H \iff \exists!$ edge between $f(u)$ and $f(v)$

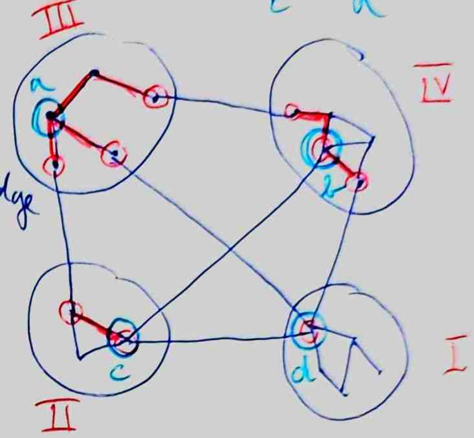
Lemma: If $\Delta(H) \le 3$, then $\forall G$: $H \le_m G \iff H \le_t G$



Proof: We need to show "$\Rightarrow$"

→ take G' and branch sets $f(u)$ as in (✱)

1. mark vertices in $f(u)$ from which there is an out edge
2. in each branch set find a $\subseteq$-minimal connected subgraph containing all marked vertices



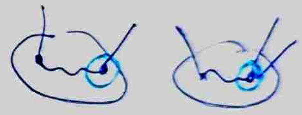
this is a tree of max degree at most 3

↳ there are at most 3 marked vertices

claim: this tree has at most 1 vertex of degree 3 ... thus we obtain a subdivision of H as a subgraph of G

and from this central vertex, the unique paths to the marked vertices are edge-disjoint

\# marked vertices: 1 ... trivial (I), 2 ... tree = shortest path (II)

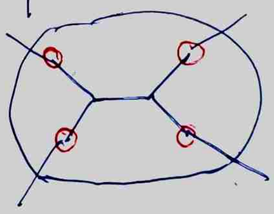


3 ... the tree has 2 or 3 leaves ...

$\blacksquare$

It no longer holds when $\Delta(H) = 4$:

↳ the central vertex may not exist



Def: A class of graphs $\mathcal{G}$ is $\preceq$-closed for some partial order $\preceq$ on graphs if

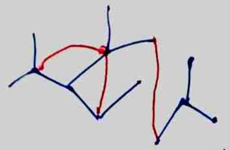
$$G \in \mathcal{G} \;\&\; H \preceq G \Rightarrow H \in \mathcal{G}$$

$\le_m$-closed $\Rightarrow$ $\le_i$-closed $\Rightarrow$ $\le$-closed but not the opposite

$\Downarrow$ $\le_t$-closed $\Rightarrow$ $\subseteq$-closed $\Rightarrow$ $\le$-closed

Problem: Which of the following graph classes are $\le_m$ or $\le_i$ closed?

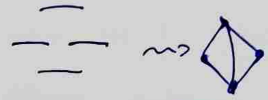
- trees ... no as deleting a vertex/edge might break connectedness
- forests ✓ ... minor of a forest is a forest
- almost k-trees = forests + $\le k$ edges ... 0-trees = forests, 3-trees:

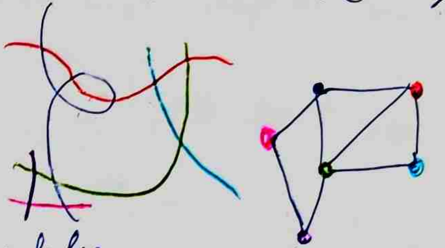


↳ equivalently: connected graphs on n vertices with at most $n+k-1$ edges and their subgraphs

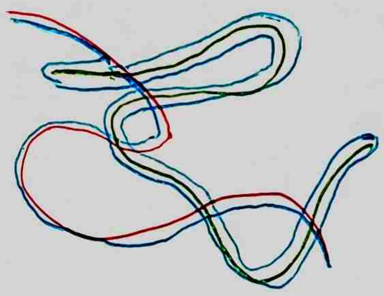
... $\le_m$ closed ✓ ... deleting vertices/edges wont make a new bad edge

... edge contraction removes 1 vertex, ≥ 1 edges

- <u>interval graphs</u> = intersection graphs of systems of $\mathbb{R}$ intervals 
  - not $\leq_m$ closed ... what if we do  → not int. graph 
  - $\leq_i$ closed ∵ deleting vertices ~ removing the interval
    contracting edge uv ~ union of intervals $I_u \cup I_v \rightsquigarrow I_{uv}$
- <u>string graphs</u> = intersection graphs of systems of planar, finite open curves (WLOG without loops) $\ell \rightsquigarrow \Omega$
  - delete vertex ~ delete curve
  - delete edge - problem
    ↳ every K_n is a string graph
    → if edge deletion would be OK, then every graph would be a string graph, but that is false



  - edge contraction



WLOG no loops $\ell \rightsquigarrow \Omega$
→ pick parent line, trace to the first intersection point, then follow the entire other line, and finish the first line

⇒ not $\leq_m$ closed, but $\leq_i$ closed

General idea: If $\mathcal{G}$ is a graph class that does not contain all graphs, but it contains every K_n, then it cannot be closed on subgraphs

Minors & graph drawing

<u>Theorem</u>: The following are equivalent

1. G is planar
2. G contains neither K_5 nor $K_{3,3}$ as a topological minor ... Kuratowski
3. —————— ‖ —————— minor ... Wagner

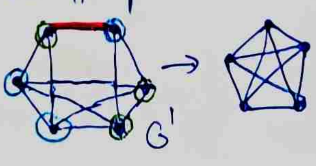
→ proved on Kombagra 2

<u>Proposition</u>: ② ⟺ ③ ... clearly "⟸"

<u>claim</u>: ¬③ ⇒ ¬② ... $K_5 \leq_m G$ or $K_{3,3} \leq_m G \Rightarrow K_5 \leq_t G$ or $K_{3,3} \leq_t G$

- if $K_{3,3} \leq_m G$, then $K_{3,3} \leq_t G$ since $\Delta(K_{3,3}) = 3$ ... lemma
- suppose $K_5 \leq_m G$ and in the series of operations there were only topo-contractions
  ↳ then $K_5 \leq_t G$

⇒ suppose the last contraction wasn't a topo contractions (we can reorder them)

$G \rightsquigarrow$ 

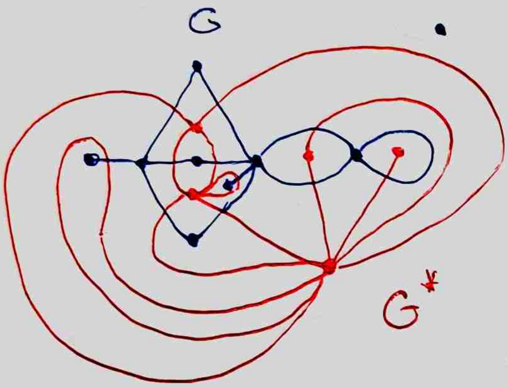
$K_{3,3} \subseteq G' \Rightarrow K_{3,3} \leq_m G$
$\Leftrightarrow K_{3,3} \leq_t G$

Note: If we want to extend the concept of minors to embeddings to a surface, then combinatorial (graph) minors and drawing minors might differ

by contracting an edge

But if we want contractions to correspond to continuous deformations of the surface, then there is no way how to obtain this drawing of H from G

Duals of planar graphs



faces of G $\rightsquigarrow$ vertices of G^*

isolated vertices of G ... ignored

vertices of degree 1 $\rightsquigarrow$ loops

edges of G ... $uv \in E$ and on one side of uv is a face h, and on the other is g (perhaps $h=g$),

$\Updownarrow$ edge between h ang g in G^*

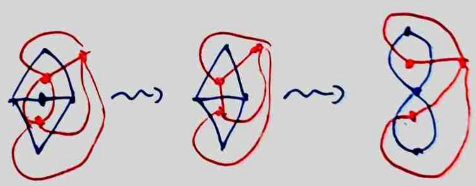
$G=(V,E) \rightsquigarrow G^*=(F,E^*)$, $e \rightsquigarrow e^*$

$G=(G^*)^*$ up to isolated vertices, since G^* ignores them

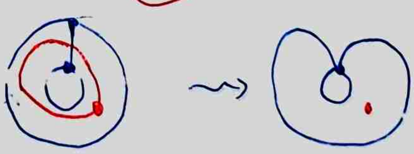
denote embeddability minors as $H \leq_e G$

Proposition: $H \leq_e G \Rightarrow H^* \leq_e G^*$

Proof:
- contraction of e $\rightsquigarrow$ deletion of e^*



- if e separates the same face:

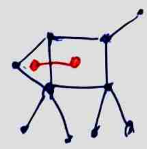


- if e is a loop, we also lose a face of G, so we need to delete the vertex in G^*
- deletion of $e \in E$ $\rightsquigarrow$ contraction of e^*, because the faces are joined
- deletion of $v \in E$ $\rightsquigarrow$ simulate by deleting adjacent edges ■

Outerplanar graphs

Def: G is outerplanar ≡ it has a planar drawing where ∀ vertex is on the outer face

Theorem: The following are equivalent



(1) G is outer planar

(2) G contains neither K_4 nor $K_{2,3}$ as a minor (or topo-minor ∵ both have $\Delta \leq 3$)

(3) G^* − vertex corresponding to the outer face is a forrest for some drawing of G
↳ delete the vertex

Proof:

(1) ⇒ (2)  … they are not outer-planar

(2) ⇒ (1) by Kuratowski-Wagner

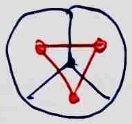
→ suppose $K_4 \not\leq_m G$ & $K_{2,3} \not\leq_m G$ and add a universal vertex v^*

→ then $K_5 \not\leq_m G+v^*$ & $K_{3,3} \not\leq_m G+v^*$, so $G+v^*$ is planar

→ take a drawing of $G+v^*$ where v^* is on the outer face (by spheric inversion)

→ remove v^* … now all $u \in G$ are on the outer face

(1) ⇒ (3) if for ∀ drawing of G, there is a cycle in $G^* - v_{outer}$ then G is not outer-planar



(3) ⇒ (1) if $G^* - v_{outer}$ is a forrest, then G is clearly outerplanar ▨

Proof of (2) ⇒ (1) without Kuratowski

→ WLOG assume that G is 2-connected; otherwise we treat the individual components of 2-connectivity separately

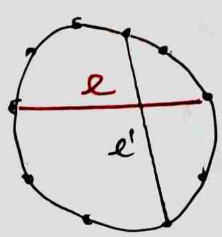
→ Ear lemma: G can be constructed from a cycle by adding "ears" < edges / paths

→ clearly ∀ cycle is outer-planar; we proceed by induction on # ears

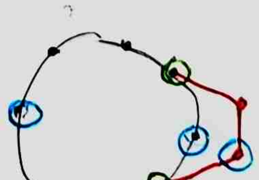
→ if G is not a cycle, find an edge e / path P
↳ by induction, G−e / G−P is outer planar

if the endpoints of P are next to each other ⇒ outerplanar drawing

a) G−e … if G not outerplanar, then e crosses some e' … BUT ! ← subdivision of K_4



b) G−P
→ otherwise:

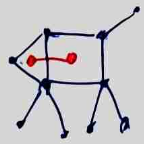


← subdivision of $K_{2,3}$! ▨

Outerplanar graphs

Def: G is outerplanar ≡ it has a planar drawing where ∀ vertex is on the outer face

Theorem: The following are equivalent



① G is outer planar

② G contains neither K_4 nor $K_{2,3}$ as a minor (or topo-minor ∵ both have $\Delta \leq 3$)

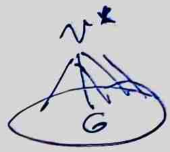
③ G^* - vertex corresponding to the outer face is a forrest for some drawing of G
↳ delete the vertex

Proof:

①⇒②  … they are not outer-planar

②⇒① by Kuratowski-Wagner

→ suppose $K_4 \not\leq_m G$ & $K_{2,3} \not\leq_m G$ and add a universal vertex v^*

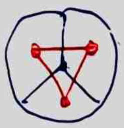


→ then $K_5 \not\leq_m G+v^*$ & $K_{3,3} \not\leq_m G+v^*$, so $G+v^*$ is planar

→ take a drawing of $G+v^*$ where v^* is on the outer face (by sphere inversion)

→ remove v^*… now all $u \in G$ are on the outer face

①⇒③ if for ∀ drawing of G, there is a cycle in $G^* - v_{outer}$
then G is not outer-planar



③⇒① if $G^* - v_{outer}$ is a forrest, then G is clearly outerplanar ▨

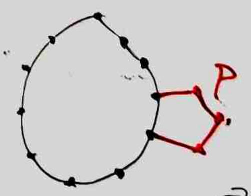
Proof of ② ⇒ ① without Kuratowski

→ WLOG assume that G is 2-connected; otherwise we treat the individual components of 2-connectivity separately

→ Ear lemma: G can be constructed from a cycle by adding "ears" < edges / paths

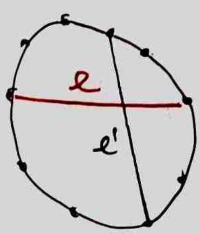
→ clearly ∀ cycle is outer-planar; we proceed by induction on # ears

→ if G is not a cycle, find an edge e / path P
↳ by induction, G−e / G−P is outerplanar



if the endpoints of P are next to each other ⇒ outerplanar drawing

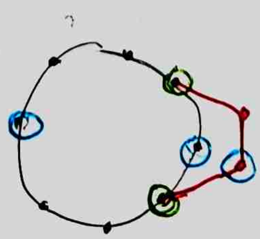
a) G−e … if G not outerplanar, then e crosses some e′



… BUT ! ← subdivision of K_4

b) G−P

→ otherwise:



← subdivision of $K_{2,3}$! ▨

INTRODUCTION TO WQO THEORY

Def: Q is quasi-order $\equiv$ reflexive & transitive

- $x \equiv y$ if $x \leq y$ & $y \leq x$
- $x < y$ if $x \leq y$ & $x \not\equiv y$

identifying $\equiv$-equiv elements produces a partial order

- x is minimal if $\forall y \leq x$ we have $y \equiv x$

Def: Q is WQO $\equiv$ $\forall$ infinite sequence $x_1, x_2, x_3, \ldots$ $\exists\, i<j$ s.t. $x_i \leq x_j$

↳ good / bad sequence

Theorem: The following are equivalent

1. Q is WQO
2. $\forall$ sequence $x_1, x_2, x_3, \ldots$ contains an infinite nondecreasing subsequence $x_{i_1} \leq x_{i_2} \leq \ldots$
3. Q admits no infinite decreasing chains an no infinite antichains (non-equiv.)
4. $\forall\, \emptyset \neq A \subseteq Q$ contains at least 1, but only finitely many min. elements

Proof: 2 $\Rightarrow$ 1, 1 $\Rightarrow$ 3, 3 $\Rightarrow$ 2 by Ramsey's theorem

3 $\Rightarrow$ 4 no infinite decreasing chains $\Rightarrow$ $\exists$ minimal element
no infinite antichains $\Rightarrow$ only finitely many

4 $\Rightarrow$ 3 by the same reasoning ∎

Remark: If κ is a regular cardinal, and Q WQO, then $\forall$ Q-sequence of length κ contains a non-decreasing subsequence of the same length

Def: A property $\varphi(x)$ is monotone $\equiv \forall x: \varphi(x)$ & $y \leq x \Rightarrow \varphi(y)$

- finitely testable $\equiv$ $\exists$ finite set $\mathcal{F}$ s.t. $\varphi(x) \Leftrightarrow (\forall F \in \mathcal{F}): F \not\leq X$

Theorem: Q WQO $\Rightarrow$ $\forall$ monotone property is finitely testable.

Proof: Define $\overline{Q} = \{x \in Q \mid \neg\varphi(x)\}$ and let $\mathcal{F}$ be non-equiv min. elements of $\overline{Q}$ $F_1, F_2, \ldots, F_m$

- if $\varphi(x)$ then $\forall F_i \not\leq X$ ... if $F_j \leq X$ then $\varphi(F_j)$ ∵ φ is monotone ↯
- if $\forall F_i \not\leq X$ then $\varphi(x)$ ... if $\neg\varphi(x)$, take $Y := \{y \in \overline{Q} \mid y \leq X\}$
  → Y has a min. element $H \leq X$, and $H \equiv F_j$ for some j ∵ H min. in $\overline{Q}$
  $\Rightarrow F_j \leq X$ ↯ ∎

Note: Graph minor thm $\Rightarrow$ $\forall$ surface $\exists$ finite obstruction set
Sphere ... $K_5, K_{3,3}$, Projective plane 35 obs., other than that not known

Preservation properties

- Q well-ordered $\Rightarrow$ Q WQO ... for example ω or $\{1,2,\dots,n\}$
- Q finite $\Rightarrow$ $(Q,=)$ WQO ... finite alphabets

Def: Product order of $P\times Q$: $(p,q)\le(p',q')$ if $p\le_P p'$ & $q\le_Q q'$

Dickson's lemma: P, Q WQOs $\Rightarrow$ $P\times Q$ WQO.

Proof: $(p_n,q_n)_n$, take ∞-nondecreasing subsequence $p_{i_1}\le p_{i_2}\le\dots$

$\Rightarrow (p_{i_n},q_{i_n})_n$, Q WQO ... $\exists\, l<k$ s.t. $q_{i_l}\le q_{i_k}$ $\blacksquare$

Def: $Q^{<\omega}$ ordered by $(a_i)_{i<n}\sqsubseteq(b_j)_{j<m}$ if $\exists$ increasing $f: n\to m$ s.t. $\forall i<n:\ a_i\le b_{f(i)}$

- $(\Sigma,=)$ finite alphabet, $u\sqsubseteq v$ is word embedding
- $P(Q)$ ordered by $X\sqsubseteq_s Y$ if $\exists$ inj. $f: X\to Y$ s.t. $\forall x\in X:\ x\le f(x)$.

Theorem (Higman, 1952): Q WQO $\Rightarrow$ $Q^{<\omega}$ WQO.

Proof: Construct a minimal bad sequence $w_0, w_1, w_2,\dots$ min $|w_i|$

- write $w_i = a_i w_i'$
- Q WQO $\Rightarrow$ $\exists$ increasing $f:\omega\to\omega$ s.t. $a_{f(0)}\le a_{f(1)}\le a_{f(2)}\le\dots$

claim: $w'_{f(0)}, w'_{f(1)}, w'_{f(2)},\dots$ is good

↳ then $\exists\, i<j$: $w'_{f(i)}\sqsubseteq w'_{f(j)}$ so $w_{f(i)}\sqsubseteq w_{f(j)}$

Proof: Suppose not; then the sequence

$$w_0, w_1,\dots, w_{f(0)-1}, w'_{f(0)}, w'_{f(1)},\dots \quad \text{is bad}$$

- if $w_j\sqsubseteq w'_{f(k)}$ for some $j<f(0)\le f(k)$ then $w_j\sqsubseteq w_{f(k)}$ ↯

Hence it is bad, contradicting the minimality of $w_{f(0)}$ since $|w'_{f(0)}|<|w_{f(0)}|$

$\blacksquare$

Kruskal's tree theorem 1960

3

Def: Order-theoretic tree is a partial order $(T, \le)$ s.t. $\forall x \in T$:

$(\leftarrow, x) := \{y \in T \mid y < x\}$ is a finite chain

There is a unique minimal element ... root

Intuition: $y < x$ means "y lies on the unique path from x to the root"

Def: A homeomorphic embeddic of two Q-labeled trees: $\varphi : S \to T$ — Write $S \trianglelefteq T$

i) $l_S(x) \le l_T(\varphi(x))$ ... respects labels

ii) $x < y \Rightarrow \varphi(x) < \varphi(y)$

iii) $\varphi(x \wedge y) = \varphi(x) \wedge \varphi(y)$

} topological minor for rooted trees

Theorem (Kruskal): Q WQO $\Rightarrow$ $T_f(Q)$ = finite Q labeled trees WQO by $\trianglelefteq$

Corollary: Finite graph-trees are WQO by topological minors... take $Q = \{\bullet\}$

By Higman's lemma also forrests.

Proof: Construct a minimal bad sequence $T_0, T_1, T_2, \dots$ minimizing $|T_i|$

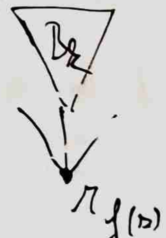
- r_i ... root of T_i
- B_i ... branch-trees of T_i

claim: $\mathcal{B} := \bigcup_{i<\omega} B_i$ is WQO by $\trianglelefteq$

proof: Suppose that $B_0, B_1, \dots$ is bad

- $\forall j$ select $f(j)$ so that $B_j \in \mathcal{B}_{f(j)}$
- let k minimize $f(k)$
- consider the sequence $T_0, T_1, \dots, T_{f(k)-1}, B_k, B_{k+1}, \dots$

$T_{f(k)}$:



it is bad, contradicting the minimality of $T_{f(k)}$ since $|B_k| < |T_{f(k)}|$

↳ if $T_i \trianglelefteq B_j$ for some $i < f(k)$ and $j \ge k$ then $T_i \trianglelefteq T_{f(j)}$ & $i < f(k) \le f(j)$

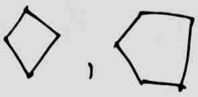
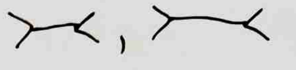
Because Q is WQO, we may assume that $\ell(r_0) \le \ell(r_1) \le \dots$

Since $\mathcal{B}$ is WQO, Higman's lemma gives $(\mathcal{B}, \sqsubseteq_s)$ WQO, so

$\exists\, i < j$ s.t. $\mathcal{B}_i \sqsubseteq_s \mathcal{B}_j$

Together, $T_i \trianglelefteq T_j$, contradicting badness of the minimal bad sequence ■

Ding's theorem 1992

👁 cycles , … and **forks** F_n  are induced subgraph $\le_i$ and subgraph $\subseteq$ antichains

Theorem (Ding) If $\mathcal{G}$ is a $\subseteq$-closed class of finite graphs, then the following are equiv:

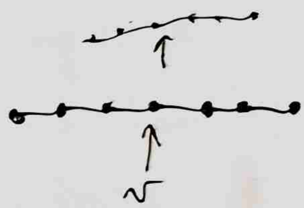
1. $\mathcal{G}$ is WQO by $\subseteq$
2. $\mathcal{G}$ is WQO by $\le_i$
3. $\mathcal{G}$ contains only finitely many C_n and F_n

clearly: ② $\Rightarrow$ ① $\Rightarrow$ ③

Theorem (Ding) If $\exists n$ s.t. no $G \in \mathcal{G}$ has $P_n \subseteq G$, then $\mathcal{G}$ is WQO by $\le_i$.

Def: Tree-depth of a graph G is

$$td(G) := \begin{cases} 1, & \text{if } |G| = 1 \\ \max_i td(G_i) & \text{if } G \text{ is disconnected with components } G_1, \dots, G_m \\ (\star)\ 1 + \min_{v \in G} td(G - v) & \text{otherwise} \end{cases}$$



👁 $td(P_n) \approx \log_2(n)$ … $td(P_n) = 1 + td(P_{\lceil n/2 \rceil})$

$\rightarrow$ if $td(G) < \ell$ then $P_{2^\ell} \not\subseteq G$ because td is minor monotone

👁 if $P_n \not\subseteq G$, then $td(G) < n$

↳ if $td(G) \ge n$, then by taking the best v in $(\star)$ we construct path $v_1, \dots, v_n \subseteq G$

Def: Induced subgraphs for Q-labeled graphs

$(G, \ell_G) \le_i (H, \ell_H)$ if $\exists$ injection $f: V_G \to V_H$ s.t.

① $uv \in E_G \iff f(u)f(v) \in E_H$ ② $\forall v: \ell_G(v) \le \ell_H(f(v))$

Notation: $D_n(Q)$ … finite Q-labeled graphs with tree-depth $\le n$

Theorem: $\forall n$: Q WQO $\Rightarrow D_n(Q)$ WQO by $\le_i$.

Proof: Induction on n … case $n=1$ holds from Higman's lemma

- let $n > 1$, sequence (G_i, ℓ_i) … WLOG $\forall i: td(G_i) \ge 2$ … there are ∞ many i
- $Q^+ := Q \times \{0,1\}$ ordered by $(q,e) \le (q',e')$ if $q \le q'$ & $e = e'$ is WQO
- $\forall i$ pick $v_i \in G_i$ s.t. $td(G_i - v_i) \le n-1$ … $H_i := G_i - v_i$
- since Q is WQO we may assume that $\ell_0(v_0) \le \ell_1(v_1) \le \dots$
- Q^+-label H_i by $\ell_i^+(u) := (\ell_i(u), e_i(u))$ where $e_i(u)$ encodes whether $uv_i \in E(G_i)$
- Induction hypothesis: $D_{n-1}(Q^+)$ is WQO $\Rightarrow \exists i < j$ s.t. $(H_i, \ell_i^+) \le_i (H_j, \ell_j^+)$
- let $\varphi: V(H_i) \to V(H_j)$ witness this, then $\varphi \cup \{(v_i, v_j)\}$ witnesses $(G_i, \ell_i) \le_i (G_j, \ell_j)$ ∎

Excursion to infinity

1954

Radó's counterexample: Higman's lemma fails for Q^ω

$Q = \{(i,j) \mid i<j<\omega\}$ ordered by $(i_1,j_1) \le (i_2,j_2)$ if $i_1 = i_2$ & $j_1 \le j_2$ or $j_1 < i_2$

→ easy to verify by Dickson's lemma that Q is WQO

	1	2	3	4	5	6	
0	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	...
1	⊛	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	...
2			(2,3)	(2,4)	(2,5)	(2,6)	...
3				(3,4)	(3,5)	(3,6)	...
4				⊛	(4,5)	(4,6)	

P(Q) is not WQO by order-respecting maps

If ⊛ ⊑ ⊛, then O ≤ something green ↯

Better-quasi-orderings ... Nash-Williams 1965

Q BQO ⇒ P(Q), $Q^{<On}$ BQO ... $Q^{<On}$ = transfinite sequences

Q BQO ⇒ $T_\omega(Q)$ = finite or infinite Q-labeled trees BQO by ≤

↳ thus the class of all graph-trees is WQO by topo-minors

Q BQO ⇒ $D_n^{\inf}(Q)$ = graphs of tree-depth ≤ n BQO by $\le_i$

Some statements do not need BQO theory

Q WQO ⇒ transfinite sequences $Q^{<On}$ with finite range WQO

Def: $S \le_h T$ if ∃ mapping $f: S \to T$ preserving immediate successors

↳ tree-homomorphism

Theorem (Smolík, 2026): One can prove without BQO theory that

T_ω = finite or infinite trees are WQO by tree-homorphisms.

Graph minor theorem

- Robertson & Seymour: finite graphs are WQO by minors
  ↳ there ∃ labeled version, also for directed graphs, multigraphs, hypergraphs...
- Thomas 1988: all graphs are not WQO by minors
  ↳ in fact, graphs of size 2^{ω} are not WQO
- Komjáth 1995: ∀ uncountable cardinal κ, are graphs of size κ not WQO

Conjecture: Countable graphs are WQO by minors

- Thomas 1989: Any class of graphs of bounded tree-width is WQO by minors.

Corollary: If H is a finite planar graph, then the class $\mathcal{G}$ of all graphs that do not contain H as a minor is WQO by minors.

GRAPH MINOR THEOREM

<u>Theorem</u> (Robertson, Seymour): The class of all finite graphs is WQO by minors.

<u>Fact</u>: "Is H a minor of G" can be done in $O(|V_G|^3)$

<u>Corollary</u>: Given any minor-closed class of graphs $\mathcal{G}$, there exists an algorithm running in $O(n^3)$ time that for any input graph G decides $G \in \mathcal{G}$.

↳ since WQO ⇒ monotone properties are finitely testable

→ problem... the obstruction set exist, but we do not know it

<u>Theorem</u> (Mohar): For $\forall$ surface S there exists a linear time alg. deciding whether G can be embedded into S.

TREE WIDTH & TREE DECOMPOSITIONS

<u>Def</u>: A <u>tree decomposition</u> of a graph G is a pair (T, X) where T is a tree and $X = \langle X_t \mid t \in V_T \rangle$ is a family of finite subsets of V_G called <u>bags</u> s.t.

i) $\bigcup X_t = V_G$ ... every vertex is in a bag

ii) $\forall e \in E_G \ \exists t \in V_T$ s.t. $e \subseteq X_t$

iii) $\forall v \in V_G$: $T \upharpoonright v := \{t \mid v \in X_t\}$ induces a <u>connected subgraph</u> in T

↳ subtree

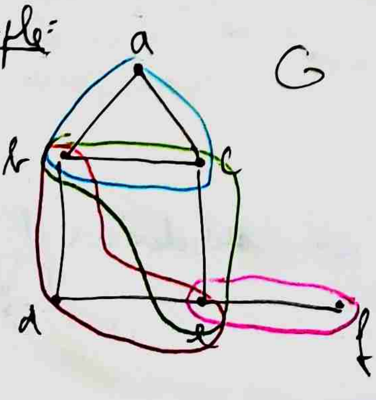
The <u>width</u> of (T, X) is $\max_{t \in V_T} |X_t| - 1$

<u>Def</u>: The <u>tree-width</u> of G is $tw(G) :=$ min width of a tree decomp of G

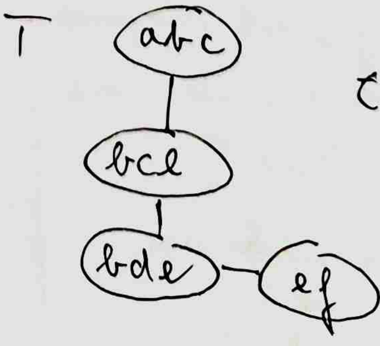
👁 iii) ⟺ If t is on the path from s to s' then $X_s \cap X_{s'} \subseteq X_t$

👁 $tw(G) \leq |V_G| - 1$ ... take a single bag with every vertex

<u>Example</u>:



⇝



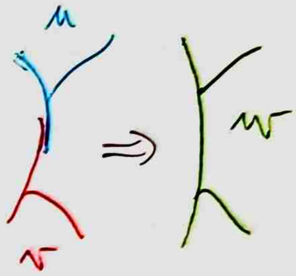
$tw(G) \leq 2$

Proposition: The following holds:

1. $H \le_m G \Rightarrow tw(H) \le tw(G)$
2. $tw(G) \ge \delta(G)$ ... min degree
3. $tw(G) \ge \chi(G) - 1$ ... chromatic number
4. $tw(G) \ge \omega(G) - 1$ ... clique number

Proof: Let (T, X) be a tree decomp. of G with min. width

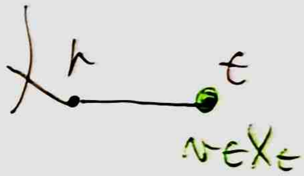
(1) H was obtained from G by
- deleting edges ... no need to change (T, X)
- deleting vertices .... remove v from bags containing it
- contracting edges ... contracting $uv \rightsquigarrow w$ $\longrightarrow$ replace u, v in bags by w



(2) If T has a single bag X_t, then $|X_t| = |V_G|$ ✓

→ otherwise let $t \in T$ be a leaf, p be the parent of t

→ WLOG $\exists v \in X_t \setminus X_p$ ... if $X_t \subseteq X_p$, we can delete t from T and try again



→ from (iii) $T \upharpoonright v = \{t\}$

⇒ from (ii) must ∀ neighbors of v appear in X_t $\Rightarrow \delta(G) \le |X_t| - 1 \le tw(G)$

(3) from (1) and (2), $\forall H \subseteq G$ contains a vertex of degree at most $tw(G)$

⇒ G is $tw(G)$-degenerated ⇒ greedily $\chi(G) \le tw(G) + 1$

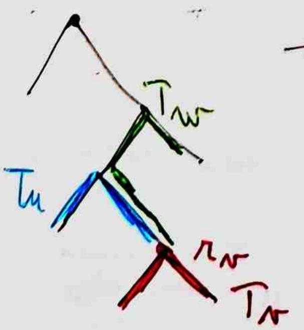
(4) claim: If $K \subseteq V_G$ induces a clique in G, then $\exists t \in V_T$ s.t. $K \subseteq X_t$

↳ letting $|K| = \omega(G)$ gives (4)

Proof: for $\forall v \in K$ put $T_v = T \upharpoonright v$. From (iii) each T_v is a subtree of T

– from (ii) $\forall u, v \in K: T_u \cap T_v \neq \emptyset$ ↗ take $t \in \bigcap$ ⊛ ?

Tree intersection lemma: If $\forall u, v: T_u \cap T_v \neq \emptyset \Rightarrow \bigcap \{T_u \mid u \in K\} \neq \emptyset$



→ root T and let T_v be s.t. its root r_v is the furthest away from the root of T

→ now $r_v \in T_u$ $\forall u \in K$

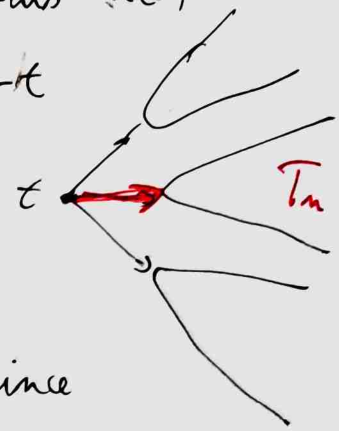
∎

Second proof of the clique claim:

→ suppose for contradiction that no X_t contains K

$\Rightarrow \forall t \in T: \exists u \in K$ s.t. $u \notin X_t$ ... choose this $u(t)$

→ remove t from T and find the component of $T-t$ containing T_u, and orient the edge →

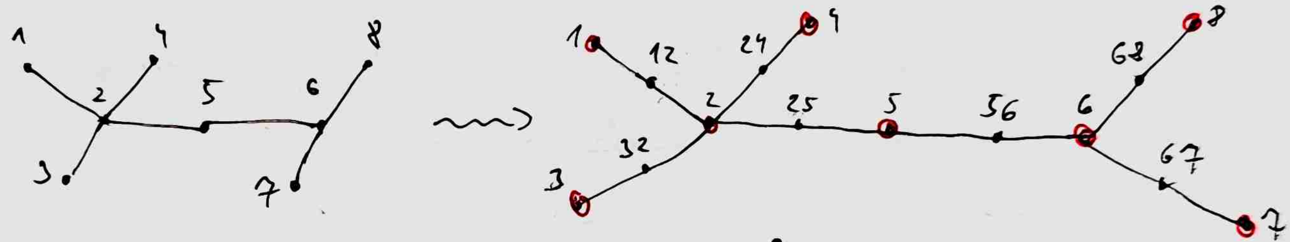


→ $|E_T| = |V_T| - 1$, so $\exists$ edge that has been oriented both directions ⟷

→ but this means that there is a cycle in T since all subtrees T_u, $u \in K$ are intersecting ▨

Observation: $tw(G) = 1 \iff G$ is a forest with at least 1 edge.

"⇐"



"⇒" if G had a cycle, it would have a $\triangle$ minor so $tw(G) \geq 2$ ▨

Lemma: If H is a subdivision of G then $tw(H) = tw(G)$

Proof: Since $G \leq_m H$ we know $tw(G) \leq tw(H)$; want $tw(H) \leq tw(G)$

→ $tw(G) = 0 \iff G$ has no edges

→ $tw(G) = 1 \Rightarrow tw(H) = 1$ since subdivision of a forest is a forest

→ if $tw(G) \geq 2$, let uv be the edge we've subdivided

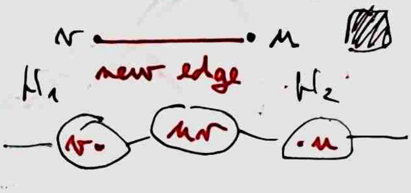
→ let (T, X) be a tree decomp. of G, we want to find a tree decomp of H with the same width

add a new bag

width has not increased

👁 $\forall H\ \exists$ connected G s.t. $H \subseteq G$ & $tw(H) = tw(G)$ ▨

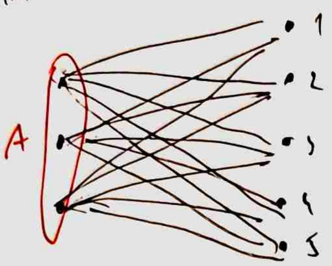
↳ H_1, H_2 connected comps, find leaf bags and add new bag

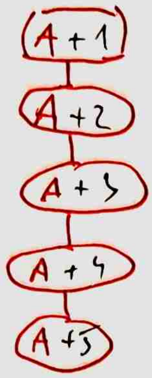


Exercise:

① $tw(K_{m,n}) = \min(m,n)$

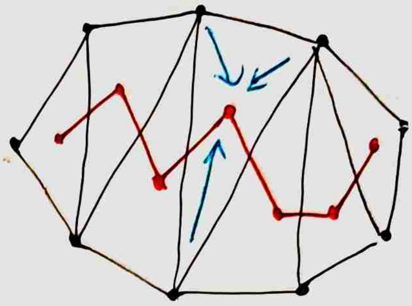
$|A| = m < n = |B|$





$tw \geq \delta = \min(m,n)$

② $tw(\text{cycle}) = 2$



③ $tw(K_n) = n-1$ ... $\omega(K_n) = n$

SERIES-PARALLEL GRAPHS

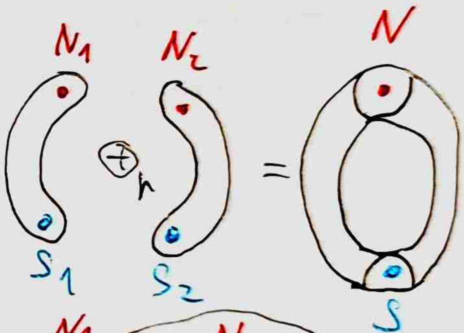
Def: Series-parallel graphs are defined recursively as follows:

- each SP graph has two special vertices – the North & the South

① $K_2 = {}^{N}_{S}$ is a SP graph

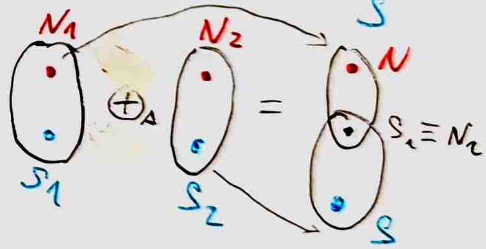
② If (H_1, N_1, S_1) and (H_2, N_2, S_2) are SP, then (G, N, S) is SP, where we combine H_1 and H_2 by identifying $N_1 \equiv N_2$ and $S_1 \equiv S_2$

parallel operation

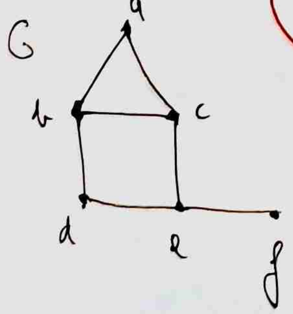


③ Or we can combine H_1 and H_2 by identifying $S_1 \equiv N_2$ and letting $N := N_1$, $S := S_2$

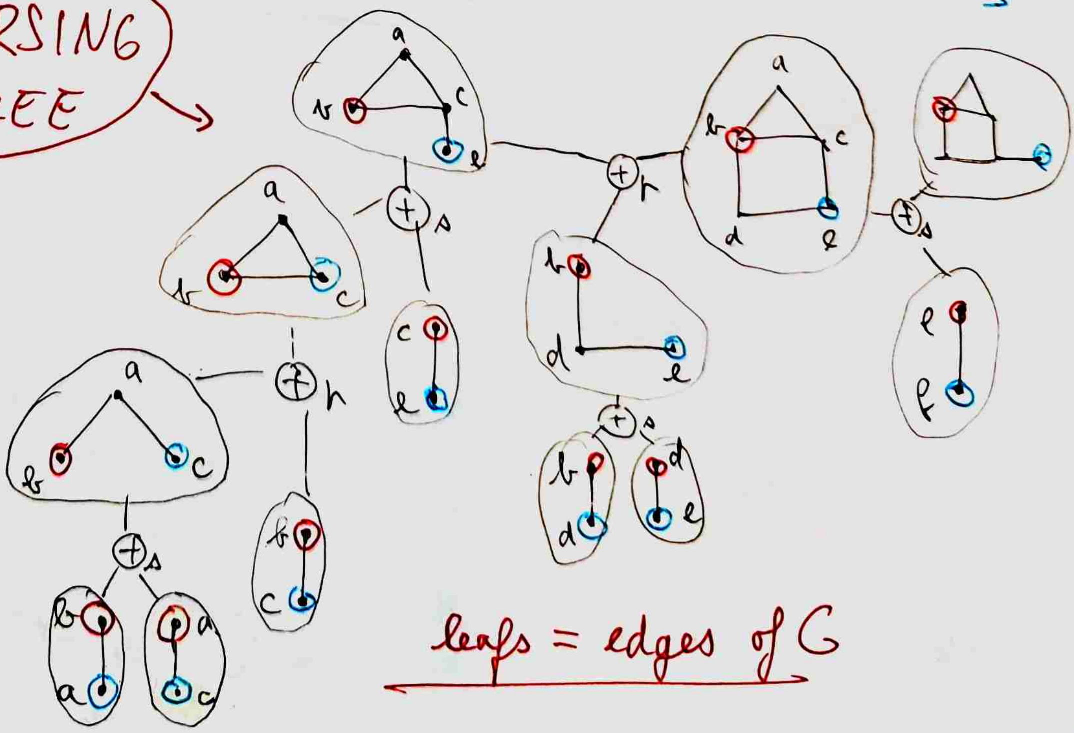
series operation



Example:



PARSING TREE



leafs = edges of G

Theorem: $tw(G) \le 2 \iff G$ is a subgraph of a SP graph

Proof: "$\Leftarrow$" we will use the parsing tree to construct a tree decomp

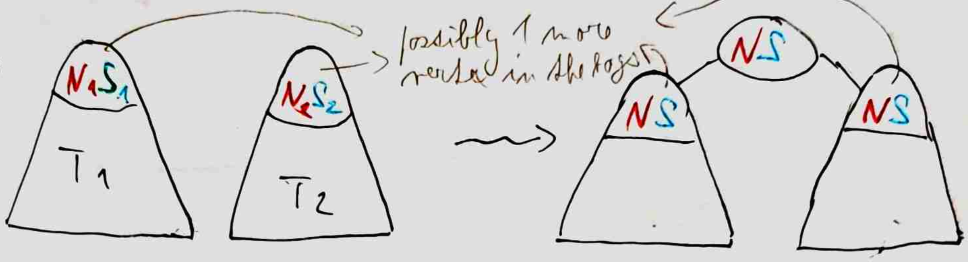
→ if H is SP, $G \subseteq H$ and $tw(H) \le 2$, then $tw(G) \le 2$

invariant: the decomp. will be rooted & the root bag will contain N and S

① K_2: N — S ⇝ T: (N, S) single bag tree

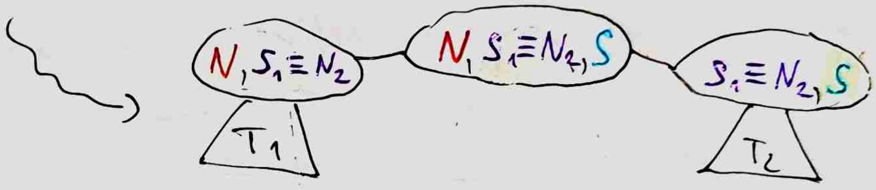
② parallel:

induction →



③ series

↳ $S_1 \equiv N_2$

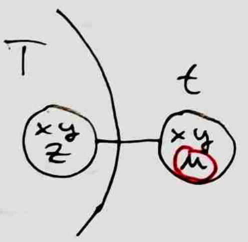


"$\Rightarrow$" Given tree decomp (T, X) of width ≤ 2, construct a SP graph H s.t. $G \subseteq H$

INDUCTION ON $|G| =: n$

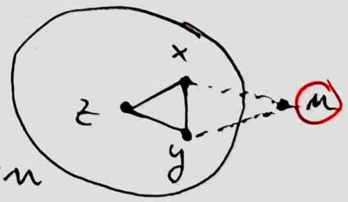
- WLOG assume that G is connected
- WLOG assume that $\forall$ bag has size 3 and induces a Δ in G
  ↳ if less, we can borrow vertices from neighbors
  → if not, we add edges to G
- WLOG every two neighboring bags have 2 vertices in common

→ let t be a leaf of T



→ X_t has exactly 1 vertex (u) that is not in the parent of t

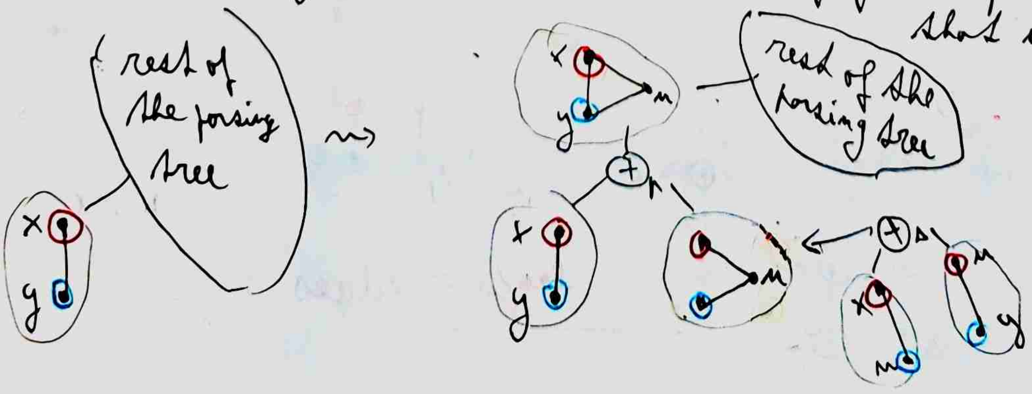
- what this looks like in G:



→ $|G-u| < |G|$ & $tw(G-u) \le 2$

⇒ by induction $\exists$ parsing tree for $G-u$

→ because $xy \in E(G-u)$, there is a leaf of the parsing tree x — y that introduced this edge



⇒ we can add in the path x — u — y to make a Δ

Theorem: G is a subgraph of a SP graph $\iff$ it has no K_4 minor

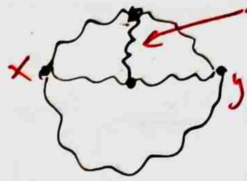
Corollary: Since G is outerplanar $\iff$ it has no K_4 or $K_{2,3}$ minor
and $tw(G) \leq 2$ $\iff$ it has no K_4 minor, we have:

$\Rightarrow$ G is outerplanar $\iff$ $tw(G) \leq 2$ & $K_{2,3} \not\leq_m G$

↳ $tw(K_{2,3}) = 2$

Proof: "$\Rightarrow$" if it had K_4 minor $\Rightarrow$ $tw \geq 3$ $\Rightarrow$ not SP

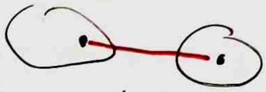
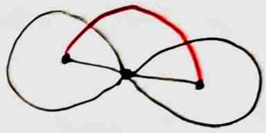
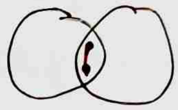
"$\Leftarrow$" $\forall$ 3-connected graph has K_4 minor



→ so we only consider graphs with connectivity ≤ 2

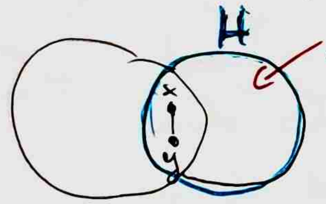
→ we again augment G:

→ we ensure that G is 2-connected without introducing a K_4 minor

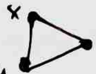
- if G disconnected, add an edge 
- if G has an articulation, add an edge between neighbors 
- if G has a 2-cut $\{x, y\}$, add xy edge 

$\Rightarrow$ now, let $\{x, y\}$ be a 2-cut of G s.t. $G - xy$ has the smallest component possible

↳ if $|G| \geq 4$ there $\exists$ 2-cut

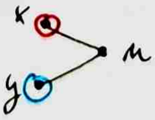


→ let H be the subgraph of G induced by this component together with x and y

H is , otherwise, if $|H| \geq 4$, then H would have a 2-cut, which would also be a 2-cut in G, contradicting the minimality of the 2-cut $\{x, y\}$

$\Rightarrow$ we can now do the same "operation" as in the previous proof

... from induction, $G - u$ has a parsing tree with $x \frown y$ leaf, and we add in



∎

TREE-WIDTH OF DUALS OF PLANAR GRAPHS

→ Recall: G outerplanar $\iff$ $\exists$ drawing of G s.t. G^* without the outer-face vertex is a forrest

Corollary: If G is outerplanar then $tw(G^*) \leq 2$. → add the outer-vertex to $\forall$ bag

Theorem (Lapoire) If G is planar then $|tw(G) - tw(G^*)| \leq 1$.

CHORDAL GRAPHS

Def: G is <u>chordal</u> $\equiv$ it contains no induced cycle of length ≥ 4.

↳ $(C_4, C_5, C_6, \dots)$-free graphs

Def: $v \in V(G)$ is <u>simplicial</u> in G $\equiv$ the neighbours of v induce a clique in G.

Theorem: The following are equivalent.

induced subgraph of a chordal graph is chordal

1. G is <u>chordal</u>
2. G has a <u>simplicial vertex</u>
3. G is the vertex <u>intersection graph</u> of a collection of subtrees in a tree

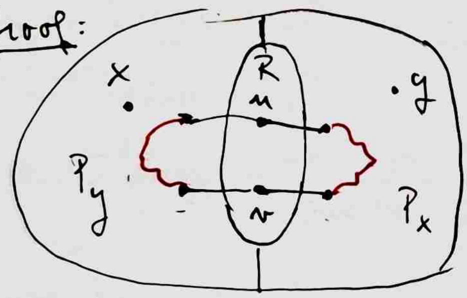
→ $\exists$ tree T, $\exists T_1, \dots, T_n$ subtrees of T $\exists$ bijection $\tau: V(G) \to \{T_1, \dots, T_n\}$

s.t. $\forall x \neq y \in V(G)$: $xy \in E(G) \iff V(\tau(x)) \cap V(\tau(y)) \neq \emptyset$

Proof: 1 ⇒ 2. If G is a clique then $\forall$ vertex is simplicial ✓

→ otherwise <u>claim</u>: If x and y are not adjacent, then $\exists$ x–y cut (separating x,y) that is a clique

<u>proof</u>:



→ let R be a smallest x–y cut and for contradiction assume that there are $u, v \in R$ that are not adjacent

→ there are edges from u and v to the components with x and y, otherwise R not minimal

paths

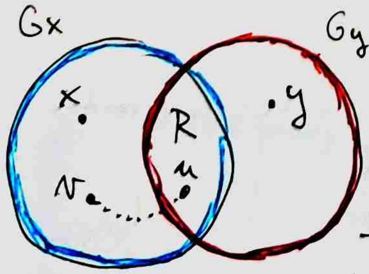
↳ let P_x and P_y connect these vertices (we are looking at a connected component)

→ take shortest paths, then they induce a cycle of len ≥ 4 in G ↯ $\boxed{}$

<u>corollary of claim</u>: by induction on $|G|$.

↳ if G is not a clique, we will find 2 nonadjacent simplicial vertices

→ let x, y be non-adjacent and R be a x–y cut that is a clique



→ let G_x = subgraph induced by R and the component of $G - R$ containing x
G_y similar

→ if G_x is a clique, x is simplicial

→ otherwise from induction $\exists u, v$ nonadjacent simplicial in G_x

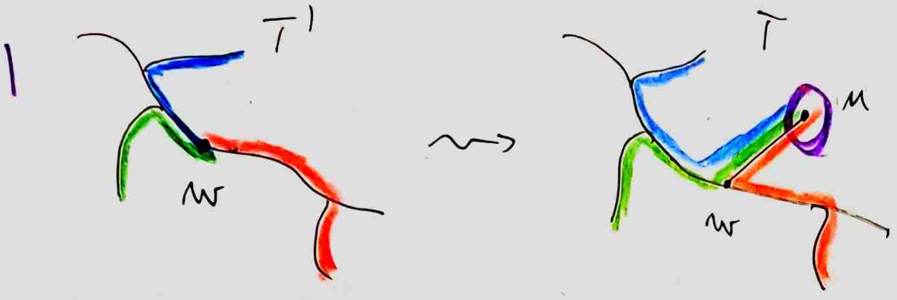
→ at least one of u, v is in $G_x - R$ $\because$ R is a clique

→ similarly we find a simplicial vertex in $G_y - R$

(2) $\Rightarrow$ (3): by induction on $|G|$, base: $|V(G)| = 1$ ✓

$n \to n+1$: let x be simplicial in G and delete it $\rightsquigarrow G' := G - x$

→ we can easily modify the trees for G' into trees for G



→ let $T'_1, \dots, T'_k$ be the subtrees of the neighbors of x

→ they all intersect so $\exists w \in T'_1 \cap \dots \cap T'_k$

→ add a new vertex u and put $T_i := T'_i \cup \{u\}$ ... add edge between u, w

→ let the subtree corresponding to x be $T_x := \{u\}$

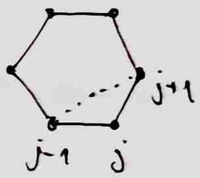
(3) $\Rightarrow$ (1): for contradiction assume that $v_1, \dots, v_k \in V(G)$ induce a ≥ 4 cycle

→ look at the corresponding trees $T_1, \dots, T_k$

→ root T, then $\forall T_i$ has an induced root r_i

→ pick the T_j with the bottom-most root

→ then the two neighbours T_{j-1} and T_{j+1} must contain r_j so they intersect, so the cycle is not induced ↯



∎

Corollary: G is chordal $\Leftrightarrow$ it has a perfect elimination scheme (PES)

→ vertices of G can be ordered as $x_1, \dots, x_n$ s.t. $\forall i$: x_i is simplicial in the subgraph induced by $x_1, \dots, x_i$

Proof: $\Rightarrow$ greedy remove simplicial verticies (G chordal $\Rightarrow$ $G-v$ chordal)

$\Leftarrow$

if there were a ≥ 4 cycle, it would not be a PES ∎

Corollary: Every chordal graph has a tree decomp s.t. $\forall$ bag induces a clique

Proof: By induction on the PES ... if we want to add a simplicial x, find $t \in T$ s.t. $N(x) \subseteq X_t$ ... exists by a previous claim $\because$ $N(x)$ is a clique

→ add new leaf node s connected to t with bag $X_s := N(x) \cup \{x\}$

👁 X_s is also a clique

∎

k-TREES

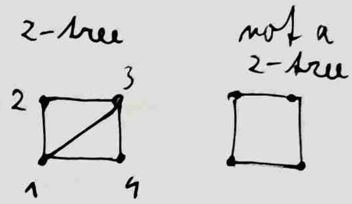
<u>Def</u>: G is a <u>k-tree</u> $\equiv$ G has a PES $x_1, \dots, x_n$ with the property that $\forall i$: x_i has $\le k$ neighbours among $x_1, \dots, x_{i-1}$

<u>Def</u>: G is a <u>partial k-tree</u> $\equiv$ it is a subgraph of a k-tree.

0-trees : • • • • = empty graphs

1-trees :  = forests

2-trees : 



<u>Observations</u>:

1. k-trees are chordal
2. k-trees are closed under induced subgraphs
3. for $k \ge 2$, a partial k-tree does not have to be a k-tree
4. partial k-trees are closed under subgraphs

<u>Theorem</u>: <u>$tw(G) \le k$</u> $\iff$ <u>G is a partial k-tree.</u>

<u>Proof</u>: $\Leftarrow$: Let $x_1, \dots, x_n$ be the PES of a k-tree H. <u>claim</u>: $tw(H) \le k$

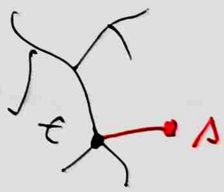
→ we inductively construct a tree decomp, starting with T = •

<u>$i \to i+1$</u>: after adding v_i let $N :=$ neighbours of v_i among $v_1, \dots, v_{i-1}$

→ since N is a clique there $\exists t$ s.t. $N \subseteq X_t$

$\Rightarrow$ add new node s connected to t with $X_s := N \cup \{v_i\}$

→ since $|X_s| \le k+1$, width $\le k$



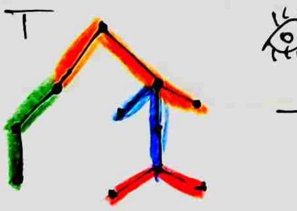
"$\Rightarrow$": Let (T, X) be a tree decomp of G of min. width

Let H be obtained from G by adding edges xy whenever $\exists t \in T$ s.t. $x, y \in X_t$

↳ we complete each bag of T into a clique

<u>claim</u>: H is a k-tree so G is a partial k-tree

↳ root T and $\forall v \in V_H$ let r_v be the induced root of $T \restriction v$



$uv \in E_H \iff T \restriction u \cap T \restriction v \neq \emptyset$

→ order the vertices of H based on how close r_v is to the root of T

<u>claim</u>: This is a PES of H

• < • ≤ • < •

→ consider the left neighbours of v_i in the PES

→ their trees have roots above / at the same level as v_i and they intersect $T \upharpoonright v_i$, so they must be included in $X_{r_{v_i}}$

→ since $X_{r_{v_i}}$ induces a clique in H, v_i is connected to all its left neighbours, and since $|X_{r_{v_i}}| \leq k+1$, there are $\leq k$ left neighbours ∎

<u>Corollary</u>: $tw(G) = \min(\omega(G')-1)$ where G' is a chordal graph obtained from G by adding edges

<u>Proof</u>: $\leq$: If G' is chordal, then it is a $(\omega(G')-1)$-tree so $tw(G') \leq \omega(G')-1$

$\geq$: we need to show that every G can be made into chordal G' without increasing tree-width. Then $tw(G) = tw(G') \geq \omega(G')-1 \geq \min$

→ but G is a partial k-tree where $k = tw(G)$

⇒ let G' be the k-tree s.t. $G \subseteq G'$ ... then still $tw(G') \leq k$ ∎

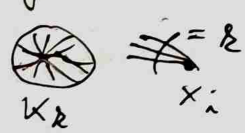
<u>Corollary</u>: Every graph G has a tree decomp with min. width and $\leq |V_G|$ nodes.

<u>Proof</u>: Let k be minimal s.t. G is a partial k-tree ... so $tw(G) = k$

→ the proof "G is a partial k-tree ⇒ $tw(G) \leq k$" constructs a decomp with $|V_G|$ nodes ∎

EXACT k-TREES

<u>Def</u>: G is an exact k-tree ≡ G has a PES $x_1, \dots, x_n$ s.t. $x_1, \dots, x_k$ form a clique & $\forall i > k$: x_i has exactly k neighbours among $x_1, \dots, x_{i-1}$



<u>Theorem</u>: $tw(G) \leq k$ ⟺ G is a subgraph of an exact k-tree.

<u>Proof</u>: Almost the same as before, but we first modify the given tree decomp. (T, X) of G of width k. We will modify it to some (T', X') of width k

→ first, add extra vertices to bags so that each bag has size exactly $k+1$ (borrow vertices from neighbours)

→ second, add extra nodes to T so that if s, t are adjacent, then $|X_s \cap X_t| = k$

$$\{1,2,3,4\} - \{1,5,6,7\} \rightsquigarrow \{1,2,3,4\} - \{1,5,3,4\} - \{1,5,6,4\} - \{1,5,6,7\}$$

→ let H be obtained from G by adding edges xy whenever $\exists t \in T'$ s.t. $x, y \in X'_t$

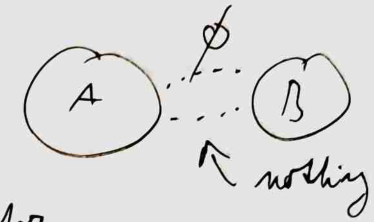
→ by essentially the same argument as previously, H is an exact k-tree ∎

SEPARATORS

Def: Let $G=(V,E)$ be a graph. If $V = A \dot\cup B \dot\cup S$ is partitioned s.t. $\forall$ path from any $a \in A$ to any $b \in B$ contains a vertex from S, we say that S separates A and B, or that S is an (A,B)-separator or cutset.

Examples:

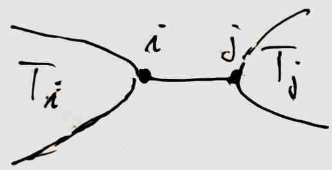
- If A and B form distinct components of G, then $\emptyset$ is an (A,B)-sep.



- If $A=\emptyset$, any S is an $(A, B=V\setminus S)$-separator

G \ S is disconnected if S separates two non-empty sets

Notation: (T,X) decomp of G



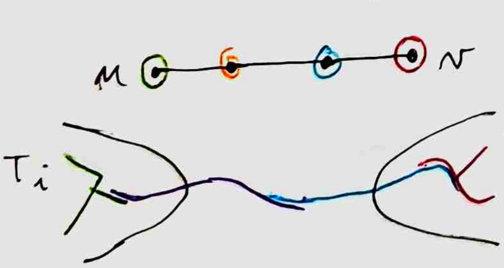
$\Rightarrow$ for $ij \in E_T$, denote by

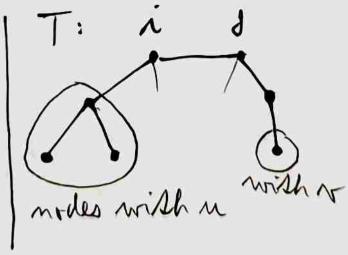
- T_i, T_j the components of $T - ij$
- G_i, G_j the subgraphs of G induced by T_i, T_j

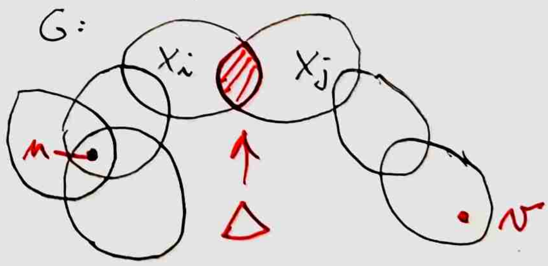
Proposition: Let (T,X) be a tree decomp. of G. Then for $\forall$ edge $ij \in E_T$, the set $\Delta := X_i \cap X_j$ separates $V(G_i)\setminus\Delta$ and $V(G_j)\setminus\Delta$.

Proof: If $u \in V(G_i)\setminus\Delta$ and $v \in V(G_j)\setminus\Delta$ were connected by a path in $G\setminus(X_i\cap X_j)$, then the vertices of this u-v path induce a subtree of T which starts in T_i and ends in T_j, so it must contain the edge ij

$\Rightarrow$ a vertex from this path must belong to $(X_i \cap X_j)$ $\lightning$ ∎







Def: Let G be a graph and $W \subseteq V_G$. We say that $S \subseteq V_G$ is a good separator for W if $\forall$ component of $G\setminus S$ contains $\leq \frac{2}{3}|W|$ vertices of W.

Def: $S \subseteq V_G$ is a good (A,B)-separator for W if it is an (A,B)-separator s.t.

$$0 < |A\cap W| \leq \tfrac{2}{3}|W| \quad \& \quad 0 < |B\cap W| \leq \tfrac{2}{3}|W|$$

$\hookrightarrow$ so S is a good separator for W

Note: While A and B are not connected to each other in $G-S$, they do not have to induce connected components.

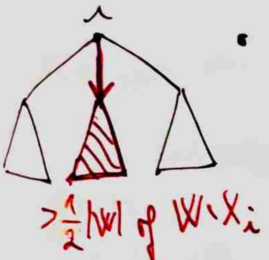
If S is a good separator for W and $|S \cap W| \geq \frac{1}{3}|W|$, then S is not necessarily an (A, B)-separator $\because$ $G-S$ might be connected

→ However, if $|S \cap W| < \frac{1}{3}|W|$ then $G-S$ has ≥ 2 components so S is always an (A, B)-separator for some A and B

Lemma: If $tw(G) \leq k$, then any $W \subseteq V_G$ has a good separator S of size $|S| \leq k+1$.

Proof: Take a tree decomp (T, X) of width k

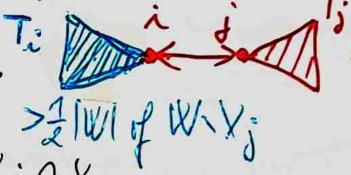
→ claim $\exists t \in T$ s.t. X_t is a good separator for W WLOG $|X_t| \geq 1 \; \forall t \in T$



- if $i \in T$ has the property that a subtree of $T-i$ contains more than $\frac{1}{2}|W|$ of $W \setminus X_i \subseteq V_G$, orient the corresponding edge from i

$>\frac{1}{2}|W|$ of $W \setminus X_i$ → at most 1 edge is oriented from each i

there is no doubly oriented edge ...



↳ $(\bigcup\{X_t \mid t \in T_i\}) \cap (\bigcup\{X_t \mid t \in T_j\}) = $ "$T_i \cap T_j$" $\subseteq X_i \cap X_j$

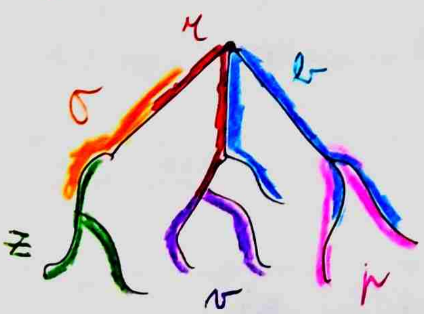
$\Rightarrow$ $T_i \cap (W \setminus X_j)$ and $T_j \cap (W \setminus X_i)$ are disjoint and both $> \frac{1}{2}|W|$ ↯

Therefore there exists a node t without an outgoin edge (we follow edges until we find one)

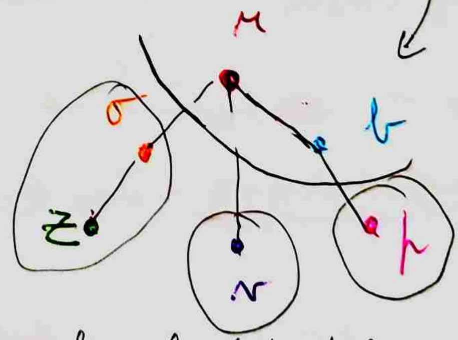
$\Rightarrow$ $\forall$ component of $G - X_t$ has $\leq \frac{1}{2}$ (and hence $\leq \frac{2}{3}$) vertices of W ∎

→ because $\forall$ component of $G-X_t$ is a subgraph of the graph induced by a subtree of $T-t$ without the vertices in X_t

! this last is an important property



⇝ $G - r - b$:



→ the subgraphs induced by $\{o, z\}$, $\{v\}$, $\{p\}$ cannot be connected together because their trees do not intersect

Lemma: If $tw(G) \leq k$, then any $W \subseteq V_G$ of size $|W| \geq 2k+3$ has a good (A,B)-separator S of size $|S| \leq k+1$.

Proof: Let (T, X) be a tree decomp of G of width k, and WLOG assume that each node has ≤ 3 neighbours:

copy bag

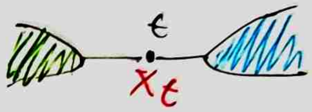
$\Rightarrow$ identify a node t as in the proof of the previous lemma

t is not a leaf since $\forall$ leaf edge is oriented away from the leaf

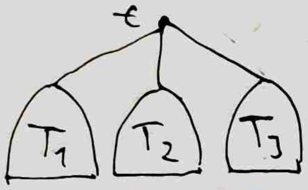
$|W \setminus X_i| \geq 2k+3-(k+1) = k+2 > \frac{1}{2}|W|$

leaf $\rightarrow$ i

a) if t has degree 2, take $S := X_t$



b) otherwise, the 3 subtrees of $T - t$ partition $W \setminus X_t$ into 3 subsets W_1, W_2, W_3



$W_i := (W \setminus X_t) \cap (\bigcup \{X_s \mid s \in T_i\})$

$\Rightarrow |W_i| \leq \frac{1}{2}|W|$ by the choice of t

$\hookrightarrow$ in particular $|W_1| \leq \frac{1}{2}|W|$

$|W_1| \geq |W_2| \geq |W_3|$ $\Leftarrow$ WLOG

$|W_2 \cup W_3| \leq \frac{2}{3}|W|$ ... otherwise $|W_2| > \frac{1}{3}|W|$ and $|W_1| < \frac{1}{3}|W|$ ↯

$\Rightarrow S := X_t$ is a good (A,B)-separator for $A \supseteq W_1$ and $B \supseteq W_2 \cup W_3$ ∎

Theorem (Reed): If every $W \subseteq V_G$ of size $|W| \geq 2k+3$ has a good (A,B)-separator of size $\leq k+1$, then $tw(G) \leq 4k+3$.

$\hookrightarrow$ so small good separators $\Leftrightarrow$ bounded tree-width

Note: Thus if $tw(G) \geq 4k+4$ there $\exists$ $W \subseteq V_G$ with no good (A,B)-sep of size $\leq k+1$.

Def: $W \subseteq V_G$ is strongly k-linked $\equiv$ it has no good (A,B) separator of size $\leq k$.

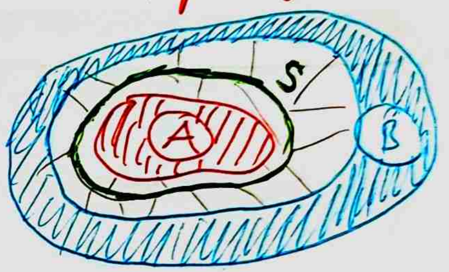
Corollary: If $tw(G) \geq 4k$, then G contains a strongly k-linked set.

Theorem (Lipton–Tarjan): Every planar graph G on n vertices contains a set $S \subseteq V_G$ of size $|S| \leq 2\sqrt{2n}$ s.t. $\forall$ component of $G \setminus S$ has $\leq \frac{2}{3}n$ vertices.
↳ so S is a good separator for $W = V_G$

Remark: $n \leq 8 \Rightarrow |S| \leq 2\sqrt{2 \cdot 8} = 8$ ... S may be V_G
$n \leq 72 \Rightarrow |S| \leq 2\sqrt{144} = 24 = \frac{72}{3}$ ... $G \setminus S$ may have a single component
$\Rightarrow$ only interesting for sufficiently large graphs

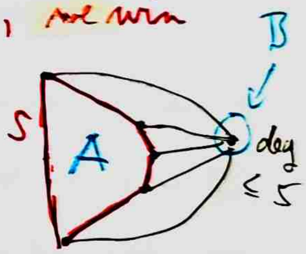
Proof: Take a planar drawing of G, make it triangulated (adding edges will not make the task easier), and if necessary, redraw it (by stereographic projection) so that there is a vertex of degree ≤ 5 on the outer face.
↳ $\exists$ because G is planar

→ consider partitions $V_G = S \dot\cup A \dot\cup B$ s.t.
S is a cycle (not necessarily induced)
with inside $= A$ & outside $= B$

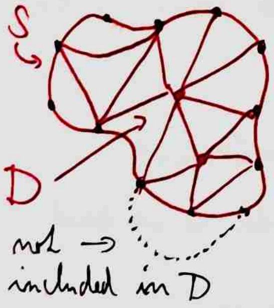


$\Rightarrow$ put $k := \lfloor \sqrt{2n} \rfloor$, and choose S such that
$|S| \leq 2k$, $|B| \leq \frac{2}{3}n$ and $|A| - |B|$ as small as possible
↳ clearly if $|A| - |B| \leq 0$, we win

↳ such a choice is possible because we can separate the vertex of deg ≤ 5 by a cycle of length ≤ 5



claim: $|A| \leq \frac{2}{3}n$ ← once we have this, we are done
→ assume for contradiction that $|A| > \frac{2}{3}n$
→ let D be the subgraph of G encircled by the drawing of S

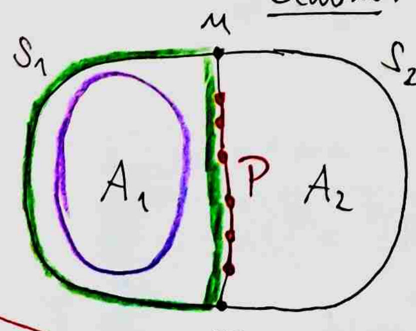


- for $u, v \in S$ denote

$c(u,v) :=$ u-v distance along the cycle S
$d(u,v) :=$ u-v distance in D
$\Big\}$ $c(u,v) \geq d(u,v)$

↖ S is the outer face of D and D is a triangulation

claim: $\forall u, v$: $c(u,v) = d(u,v)$



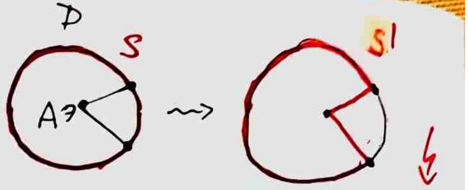
→ assume $\exists u, v$ s.t. $d(u,v) < c(u,v)$
↳ let P be the internal vertices of a u-v shortcut
$\Rightarrow$ interior $A = A_1 \dot\cup P \dot\cup A_2$, WLOG $|A_1| \geq |A_2|$
→ cycle $S = S_1 \dot\cup \{u, v\} \dot\cup S_2$
$\Rightarrow$ then $A' := A_1$, $S' := S_1 \dot\cup P \dot\cup \{u, v\}$, $B' := B \dot\cup A_2$ is a better 3-partition than A, S, B ↯

<u>continuation of proof of claim:</u> because $|P| \le |S_1|$, $|P| \le |S_2|$ since $d(a,r) < c(a,r)$

- $|S'| = |S_1| + 2 + |P| \le |S_1| + 2 + |S_2| = |S| \le 2k$
- $|B'| = n - (|A'| + |S'|) \le n - \frac{n}{3} = \frac{2}{3}n$

  ↳ $|A'| + |S'| \ge |A_1| + |P| \overset{\bullet}{\ge} \frac{1}{2}|A| \ge \frac{n}{3}$ $\qquad \bullet\ |A_1| \ge |A_2|$
- $|A'| - |B'| < |A| - |B|$, contradicting the minimality of A, S, B

→ we now know that $\forall u, v: \; c(u,v) = d(u,v)$

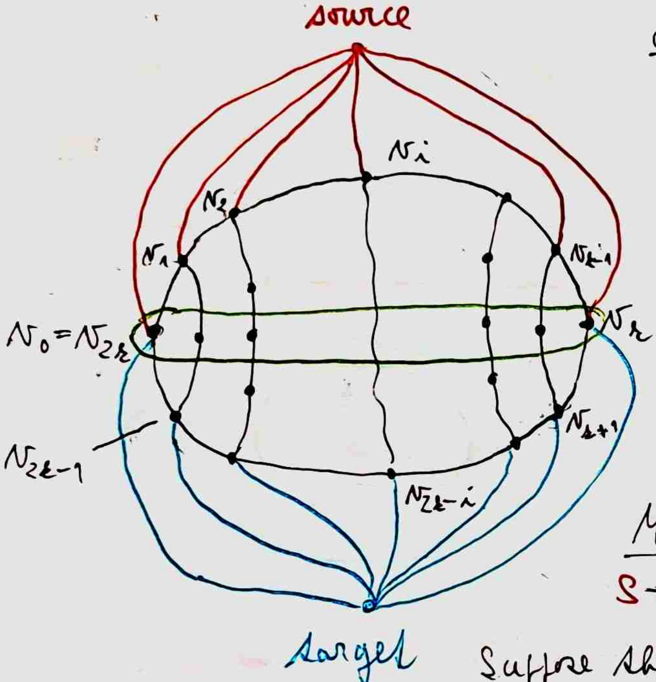


we can make better (A', S', D)

👁 $|S| = 2k$ ... if $|S| < 2k$, consider an arbitrary pair of consecutive vertices of S and the Δ face of D containing them

→ the third vertex of Δ must be in A, otherwise it would contradict $c(u,v) = d(u,v)$ ⟶

Let $v_0, v_1, \dots, v_{2k} = v_0$ be the vertices of S



claim: There exist vertex disjoint paths $P_0, P_1, \dots, P_k$ in D s.t. P_i has endpoints v_i and v_{2k-i}

→ note that $P_0: v_0 - v_{2k}$ and $P_k: v_k - v_k$ have length 0.

proof: add two new vertices: Source & Target

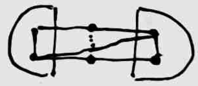
- attach S to $v_0, v_1, \dots, v_k$
- attach T to $v_k, v_{k+1}, \dots, v_{2k}$

Menger's theorem: There $\exists\, k+1$ vertex disjoint S–T paths $\iff$ there is no S–T cut of size $\le k$.

Suppose there is a cut of size $\le k$

→ since there are paths $S - v_0 - T$ and $S - v_k - T$, the cut must contain v_0 and v_k

👁 cut in planar triangulation is connected ...



→ if the cut edge were missing, then $\exists$ cross-edge ↯ ∵ triangulation

$\Rightarrow$ the cut contains a $v_0 - v_k$ path of len $< k = c(v_0, v_k)$ ↯

Therefore there $\exists\, k+1$ vertex disjoint S–T paths

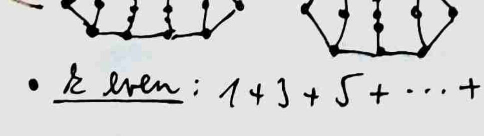
- their start vertices are $v_0, v_1, \dots, v_k$ and end vertices are $v_k, v_{k+1}, \dots, v_{2k}$

↳ how do we know they are paired up as $v_i - v_{2k-i}$?

↳ because it is a planar drawing, otherwise the paths would intersect

Because of the $c(u,v) = d(u,v)$ constraint, total # vertices on paths is at least

- k odd: $1 + 3 + 5 + \dots + k + k + \dots + 5 + 3 + 1 = 2\left(\frac{k+1}{2}\right)^2 = \frac{1}{2}(k+1)^2$



- k even: $1 + 3 + 5 + \dots + (k-1) + (k+1) + (k-1) + \dots + 5 + 3 + 1 = k+1 + 2\left(\frac{k}{2}\right)^2 > \frac{1}{2}(k+1)^2$

$\Rightarrow$ in both cases we have $|V(D)| \ge \frac{1}{2}(k+1)^2 = \frac{1}{2}\left(\lfloor \sqrt{2n} \rfloor + 1\right)^2 > \frac{1}{2} 2n = n$ ↯

↳ more vertices than the entire graph

∎

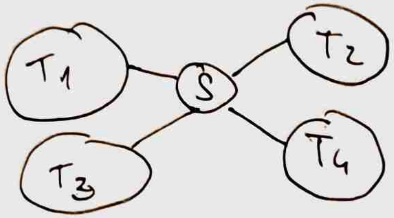
Corollary: If G is planar, then $tw(G) \in O(\sqrt{n})$ $\quad n = |V_G|$

Proof: We recursively find small separators using the Lipton–Tarjan theorem

→ let S be a separator s.t. $|S| \le 2\sqrt{2n}$ and $\forall$ component C_i of $G \setminus S$: $|C_i| \le \frac{2}{3}n$

⇒ recursively get tree decompositions for each component C_i

⇒ combine them



add a new bag $X_t = S$ and connect it to each component

→ moreover, add the vertices in S to every bag

Denote by $T(n)$ the max size of a bag in a tree decomp of a planar graph of size n that was constructed this way

$$T(n) \le |S| + \max_i T(|C_i|) \le 2\sqrt{2n} + T\left(\tfrac{2}{3}n\right)$$

$$\le 2\sqrt{2n} + 2\sqrt{2 \cdot \tfrac{2}{3}n} + 2\sqrt{2 \cdot \tfrac{4}{9}n} + \cdots$$

$$\le 2\sqrt{2n}\left(1 + \sqrt{\tfrac{2}{3}} + \left(\sqrt{\tfrac{2}{3}}\right)^2 + \cdots\right) \le 2\sqrt{2n}\,\frac{1}{1-\sqrt{2/3}} < 16\sqrt{n}$$

We will soon see that the planar $n \times n$ grid has $tw = n$, so in fact:

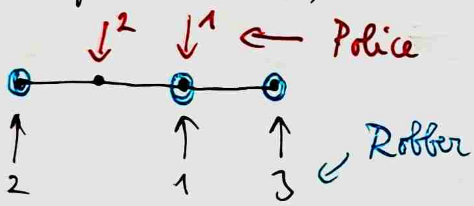
$tw(\text{planar graphs}) \in \Theta(\sqrt{n})$

SEARCH GAMES & BRAMBLES

Problem: police and robber are playing a game on a $\underbrace{\text{graph}}_{}$ (connected) — vertices = crossings, edges = streets

- *initial position:* robber occupies a vertex and police has k helicopters outside of the graph G
- *one round:*
  1) police player lifts 1 helicopter to the air and announces on which vertex it will land later
  2) robber can move to any vertex of G reachable without crossing a vertex that contains a helicopter
  3) the helicopter lifted by police lands
- *goal:*
  - *police:* catch the robber: land a helicopter on his vertex
  - *robber:* escape indefenitely

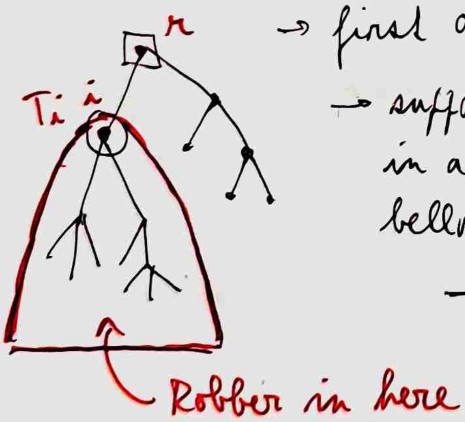
Example: $G = P_4$, $k = 1$



With 1 helicopter, this game can clearly continue forever
$\Rightarrow$ but police can clearly win with 2 helicopters

Proposition: If $tw(G) \le k$, then police wins with $k+1$ helicopters.

Proof: Root a tree decomp (T, X) of G of width $\le k$



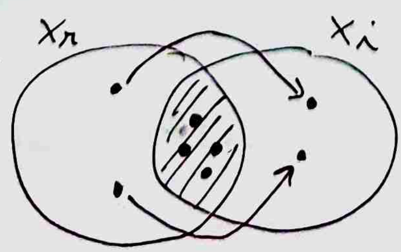
→ first occupy the entire root — $k+1$ helicopter's there

→ suppose that the robber is at a vertex v that is in a bag of some $j \in T$ that is inside a subtree T_i bellow the root r

$\Rightarrow$ we want to move the helicopters to X_i without making it possible for the robber to escape back through r

Recall: if $i - r$ is an edge in T, then $X_i \cap X_r$ separates $G_i - X_r$ from the rest.

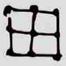
graph induced by T_i



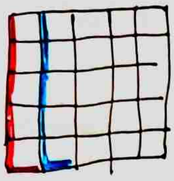
$\Rightarrow$ we can move the helicopters from $X_r \setminus X_i$ to $X_i \setminus X_r$ one by one, while keeping $X_i \cap X_r$ in place

$\Rightarrow$ like this, we drive Robber to a leaf bag ∎

What about the planar grid?

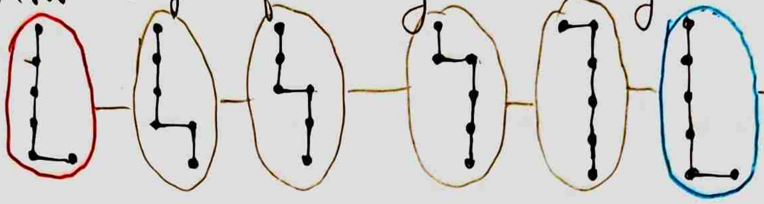
Notation: $\boxplus_k := k \times k$ grid ... so  $= \boxplus_3$

$tw(\boxplus_k) \leq k$

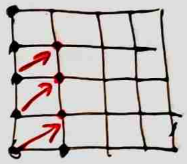


→ the tree will be a path going left → right

How to get from bag **L** to bag **L**:



Therefore police wins with $k+1$ helicopters as follows:



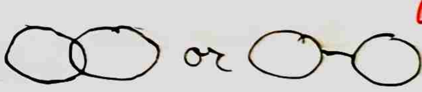
Proposition: If police has a winning strategy with $k+1$ helicopters then $tw(G) \leq k$.

↳ we will prove this using brambles which provide a strategy for the Robber

BRAMBLES

Def: A family of subsets $\mathcal{B} \subseteq P(V_G)$ is a bramble of G if

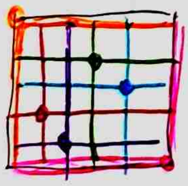
1. each $B \in \mathcal{B}$ induces a connected subgraph in G
2. $\forall B, B' \in \mathcal{B}$: $B \cup B'$ induce a —— " ——, so B and B' touch

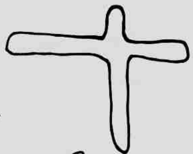
Def: We say that $W \subseteq V_G$ covers $\mathcal{B}$ if $\forall B: W \cap B \neq \emptyset$. 

Def: The order of a bramble $\mathcal{B}$ is the size of its smallest cover.

Example: Find the best (highest order) bramble for $\boxplus_k$.

order $= k$



$B_{ij} =$ 

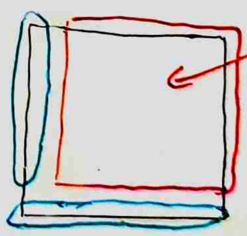
$|\mathcal{B}| = k^2$

crosses

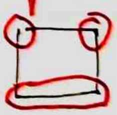
k vertices are enough to cover (take any row)

$k-1$ not enough ... then $\exists$ uncovered row & column

order $= k+1$ ... modify it a bit



crosses for $k-1$

$\boxplus_2$: 

$\boxplus_3$: 

Proposition: If G has a bramble of order $k+1$, then $tw(G) \geq k$.

Corollary: $tw(\boxplus_k) = k$.

Proof: Bramble $\mathcal{B}$ of order $k+1$ provides a winning strategy for Robber against k helicopters ... then if $tw(G) \leq k-1$, police wins with k helicopters ↯

→ Robber can always move from any $B \in \mathcal{B}$ to $B' \in \mathcal{B}$ because they touch

→ the police has only k helicopters → cannot cover every $B \in \mathcal{B}$ ▨

Proof #2:

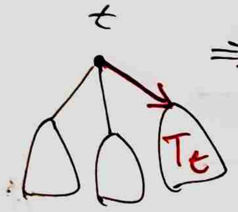
<u>claim</u>: If $\mathcal{B}$ is a bramble in G and (T, X) a tree decomp of G, then $\exists t \in T$ s.t. X_t covers $\mathcal{B}$.

<u>proof</u>: For contradiction assume that $\forall t \in T$, X_t does not cover $\mathcal{B}$

→ let $B_t \in \mathcal{B}$ be not intersected by X_t

⇒ since B_t is connected, its vertices induce a connected subtree of T

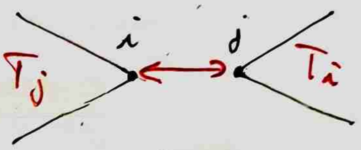
↳ this subtree T_t does not contain t since $B_t \cap X_t = \emptyset$



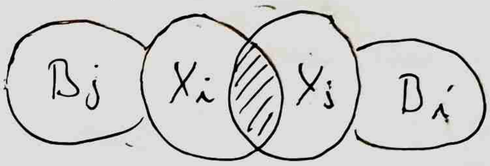
⇒ identify unique edge from t to T_t and orient it towards T_t

- $|T|$ vertices & $|T|-1$ edges

⇒ some edge is doubly oriented



👁 the cutset $X_i \cap X_j$ separates B_i from B_j so they do not touch ↯



■

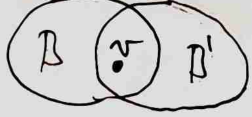
Proof #3:

→ for $\forall B \in \mathcal{B}$ put $T \restriction B := \{t \in T \mid X_t \cap B \neq \emptyset\}$

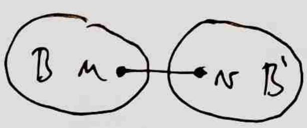
<u>claim</u>: $\forall B, B' \in \mathcal{B}$: $T \restriction B \cap T \restriction B' \neq \emptyset$

↳ since every pair of trees intersects, they all intersect ⇒ $\exists t: X_t \cap B \neq \emptyset \ \forall B \in \mathcal{B}$. ↱ X_t covers $\mathcal{B}$

<u>proof</u>: two cases of how B and B' touch

a) 

then $T \restriction B$ and $T \restriction B'$ share the entire subtree of v

b) 

then $T \restriction B$ and $T \restriction B'$ share the node with the bag containing the edge uv ■

THE TREE-WIDTH DUALITY THEOREM

<u>Theorem</u> (<u>Seymour, Thomas</u>): <u>$tw(G) \geq k$</u> $\iff$ G contains a <u>bramble of order $k+1$</u>.

<u>Corollary</u>: $tw(G) \leq k \iff$ police wins with $k+1$ helicopters

<u>Proof</u>: Already know "⟹". Suppose police wins with $k+1$ and $tw(G) \geq k+1$. Then G has a bramble of order $k+2$, which gives the robber a winning strategy against $k+1$ helicopters ↯ ■

<u>Remark</u>: Also <u>min-max</u> theorem: $\max\limits_{\mathcal{B}} \min\limits_{W} |W| = \min\limits_{T} \max\limits_{X} |X|$

bramble ↗ cover ↰ tree decomp ↱ bag ↖

Proof of the theorem: We need "$\Longrightarrow$", we will prove something more general

Lemma: If G has no bramble of order $k+1$, then $\forall$ bramble $\mathcal{B}$ $\exists$ tree decomp T s.t.
$\forall t \in T$: if $|X_t| \geq k+1$, then t is a leaf & X_t does not cover $\mathcal{B}$.

Lemma $\Rightarrow$ Theorem:

→ suppose that G has $tw \geq k$ but no bramble of order $k+1$. Then apply the lemma on the bramble $\mathcal{B} := \{V_G\}$. Since $\mathcal{B}$ is covered by any $\emptyset \neq W \subseteq V_G$, there $\exists T$ s.t. $\forall t$: $|X_t| \leq k$, so $tw(G) < k$ ↯ ■

Proof of lemma: Fix G that has no bramble of order $k+1$

→ order all brambles of G by the number of sets $W \subseteq V_G$ of size $\leq k$ covering them
↳ so $\mathcal{B} = \{V_G\}$ is last ∵ everything covers it

WLOG: $|V_G| \geq k+1$
↳ otherwise take $T = (V_G)$ ← single node with everything

→ we process all brambles in this order

- given $\mathcal{B}$, choose W smallest possible cover of $\mathcal{B}$ ... $|W| \leq k$
- let $A_1, \dots, A_r$ be the components of $G \setminus W$ ... $r \geq 1$

claim: $\forall i = 1, \dots, r$ the graph $G_i := G[A_i \cup W]$ has a tree decomp T_i s.t.
① W is a bag of T_i ② $\forall t \in T_i$ s.t. $|X_t| \geq k+1$ is a leaf & X_t does not cover $\mathcal{B}$

case A: $\mathcal{B} \cup \{A_i\}$ is not a bramble

put $T_i :=$ (s) $X_s = W$ ← $|X_s| \leq k$
— (t) $X_t = A_i \cup N(A_i)$

$A_i \cup N(A_i)$ does not cover $\mathcal{B}$
↳ since $\mathcal{B} \cup \{A_i\}$ is not a bramble, $\exists B \in \mathcal{B}$ s.t. B does not touch A_i, so $N(A_i) \cap B = \emptyset$

case B: $\mathcal{B}' := \mathcal{B} \cup \{A_i\}$ is a bramble

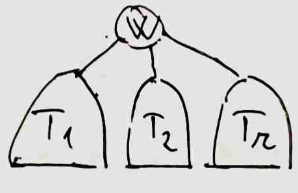
$\forall$ set that covers $\mathcal{B}'$ also covers $\mathcal{B}$, but W does not cover $\mathcal{B}'$ as $W \cap A_i = \emptyset$
⇒ $\mathcal{B}'$ has strictly less covers than $\mathcal{B}$, so we already processed it

- by induction $\exists$ tree decomp T' of G w.r.t. $\mathcal{B}'$ ... $\forall t \in T'$: $|X_t| \geq k+1 \Rightarrow t$ is a leaf & X_t does not cover $\mathcal{B}'$
↳ if T' is valid for $\mathcal{B}$ as well, take it
→ otherwise $\exists t \in T'$ s.t. $|X_t| \geq k+1$ and t is a leaf but X_t covers $\mathcal{B}$

Idea: We will create T_i from T' so that it satisfies the claim

→ here we get lucky and can move on to another bramble.

→ if we always get unlucky in case B and get all trees $T_1, T_2, \dots, T_r$, we create the final tree decomp T for $\mathcal{B}$ by gluing together $T_1, \dots, T_r$ using their shared bag W

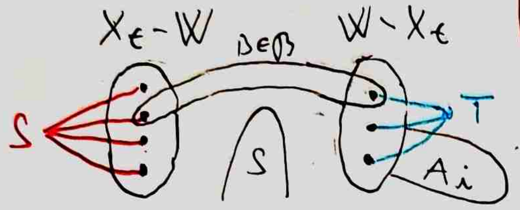


How do we get T_i from T'? $|X_\epsilon| \geq |W|$ ∵ W is smallest cover of $\mathcal{B}$

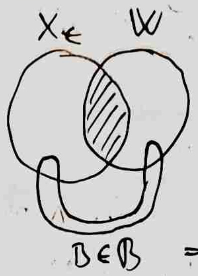
we have: $\epsilon \in T'$ s.t. $|X_\epsilon| \geq k+1$ is a leaf that does not cover $\mathcal{B}'$ but does $\mathcal{B}$

$A_i \cap X_\epsilon = \emptyset$ .... otherwise X_ϵ would cover $\mathcal{B}' = \mathcal{B} \cup \{A_i\}$

- If $W \subseteq X_\epsilon$: go down to modify T' into T_i by deleting vertices outside G_i
- otherwise: Add Source to $X_\epsilon - W$ and Target to $W \setminus X_\epsilon$



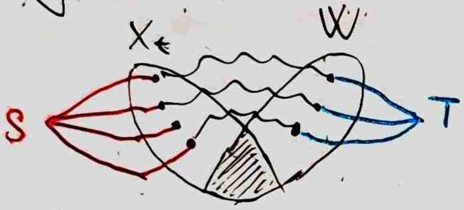
claim: Every S–T cutset S has size $|S| \geq |W - X_\epsilon|$



→ because $|W|$ is minimal, $\forall w \in W \setminus X_\epsilon$ is needed for some $B \in \mathcal{B}$. Since B were not intersect by $W \cap X_\epsilon$ and X_ϵ covers $\mathcal{B}$, there $\exists x \in X_\epsilon \setminus W$ s.t. $x \in B$.

⇒ S must intersect all of these B so $|S| \geq |W - X_\epsilon|$ otherwise $(W \cap X_\epsilon) \cup S$ is better than W ↯

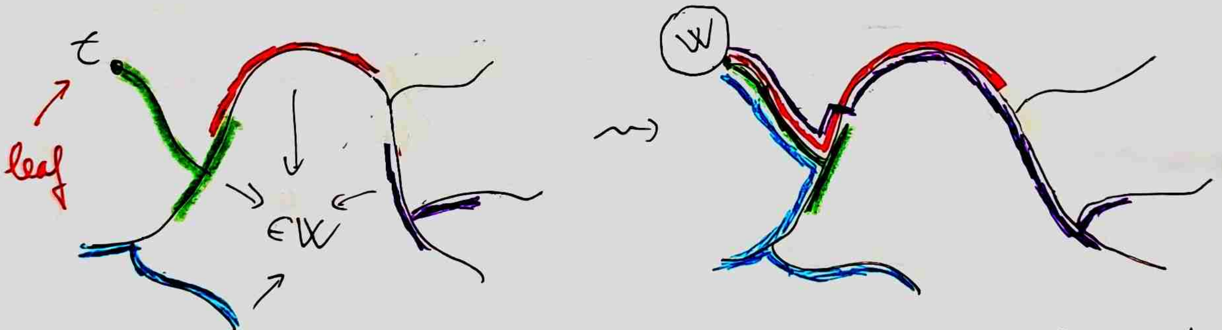
(*) Menger: $\exists$ vertex disjoint paths from T to S



→ We now modify T' into T_i as follows:

(1) restrict all bags to $V(G_i) = A_i \dot\cup W$

(2) $\forall w \in W$: insert w to every bag X_s s.t. $s \in T'$ is on a path from ϵ to $T \restriction w$ ← if $W \subseteq X_\epsilon$, this does nothing



- after the changes: $X_\epsilon \rightsquigarrow W$, so size $\leq k$ (since $A_i \cap X_\epsilon = \emptyset$)
- nothing was added to any other leaf so if a leaf bag has size $\geq k+1$ after the changes, it had before as well ... $|X_s| \geq k+1$
  ⇒ so X_s does not cover $\mathcal{B}' = \mathcal{B} \cup \{A_i\}$
  → since we restricted to G_i we have $X_s \subseteq W \dot\cup A_i$
  ⇒ $X_s \cap A_i \neq \emptyset$ as $|W| \leq k$ so X_s does not intersect some $B \in \mathcal{B}$

} no leaf of size $\geq k+1$ covers $\mathcal{B}$

- we need to verify that extending the subtrees of $w \in W$ towards ϵ did not increase the bag size of internal vertices (since we want bag $\geq k+1$ ⇒ leaf)

→ if $w \in X_\epsilon$ originally, we added w nowhere

→ otherwise, the addition of $w \in W \setminus X_\epsilon$ is compensated by the deletion of a vertex on the $X_\epsilon - W$ path (*), which is disjoint from W and A_i ∎

THE GRID MINOR THEOREM

Theorem (Robertson, Seymour): $(\forall n)(\exists k)$ $tw(G) \geq k \Rightarrow ⊞_n \leq_m G$. minor

Corollary: For every minor-closed graph class $\mathcal{G}$, the following are equiv.:

1. $\mathcal{G}$ has unbounded tree-width ... $\forall k\ \exists G \in \mathcal{G}: tw(G) \geq k$
2. $\mathcal{G}$ contains all planar graphs
3. $\mathcal{G}$ contains all grids $⊞_n$

Proof: ① $\Leftrightarrow$ ③ by the theorem & using $tw(⊞_n) = n$

② $\Rightarrow$ ③ ... since $⊞_n$ is planar

③ $\Rightarrow$ ② follows from the fact that $\forall$ planar G $\exists n$: $G \leq_m ⊞_n$

↳ imagine the infinite square grid $⊞_\infty$

↳ intuitively, one can "zoom out" and map the drawing of G onto $⊞_\infty$ to create an approximation of the drawing

→ vertices $v \in V(G)$ ⇝ "blobs" = disjoint connected subgraphs of $⊞_\infty$

→ edges $e \in E(G)$ ⇝ disjoint paths between these blobs

⇒ finally take the smallest $⊞_n$ enclosing this, delete everything unused, contract blobs and contract paths ∎

Corollary: The following are equiv for any graph H.

1. H is planar
2. the class $\mathcal{G}(H)$ of graphs without an H-minor has bounded tree-width

Proof: ① $\Rightarrow$ ②, if $\mathcal{G}(H)$ had unbounded tw then $H \in \mathcal{G}(H)$ ↯

$\neg$① $\Rightarrow \neg$② if $\mathcal{G}(H)$ had bounded tw then there would be some $⊞_n \notin \mathcal{G}(H)$ meaning that $H \leq_m ⊞_m$ but H is not planar ↯ ∎

→ we will prove a weaker version of the grid-minor theorem:

Theorem: If G is planar and $tw(G) \geq 6k-4$, then $⊞_k \leq_m G$. Robertson Seymour Thomas

→ we will show this with $tw(G) \geq 20k-4$

Recall: If $tw(G) \geq 4k$, then G contains a strongly k-linked set $W \subseteq V_G$.

↳ meaning that W has no good (A, B)-separator of size $\leq k$

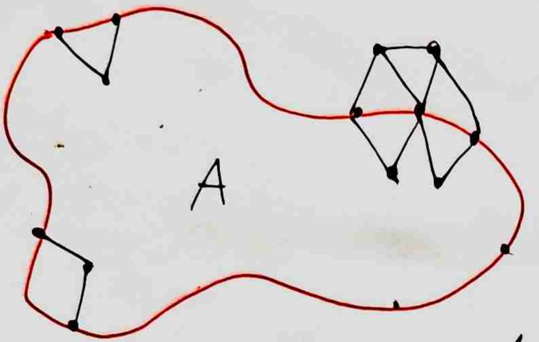
↳ so $\forall S \subseteq V_G$ of size $|S| \leq k$, some component A of $G \setminus S$ contains $> \frac{2}{3}|W|$ vertices of W

Proof: Let G be a planar graph with $tw(G) \geq 20k-4$

↳ so G contains a strongly $(5k-1)$-linked set W

$\Rightarrow$ hence $\forall S \subseteq V_G$ s.t. $|S| \leq 4k$, some component A of $G \setminus S$ contains $> \frac{2}{3}|W|$ vertices of W

- Fix a planar drawing of G and represent each such S by a closed simple (non-intersecting) curve $\mathcal{C}$ s.t.



① $\mathcal{C}$ intersects the drawing of G only in vertices

② # vertices it contains $\leq 4k$

③ int $\mathcal{C}$ contains $> \frac{2}{3}|W|$ vertices of W

→ such $\mathcal{C}$ exists as we trap A inside and route $\mathcal{C}$ through the neighbours of A as all of them have to be in S and $|S| \leq 4k$

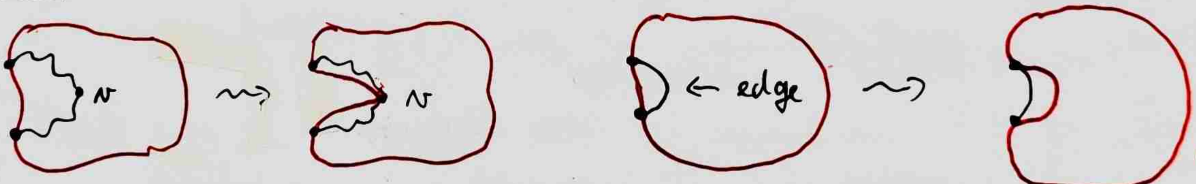
- Choose S so that int $\mathcal{C}$ contains as few vertices as possible

claim: $|S| = 4 \cdot k$ ... $\mathcal{C}$ contains precisely $4k$ vertices

proof: Suppose $|S| < 4k$, we will make S' s.t. int $\mathcal{C}'$ contains less vertices than int $\mathcal{C}$:

→ take any two consecutive vertices on $\mathcal{C}$ and look at the face they share through which $\mathcal{C}$ is routed → face $F \subseteq V_G$

case A: there $\exists v \in F$ inside int $\mathcal{C}$. case B: no such $v \in F$ exists



NOW GOTO CASE A)

→ we decreased # vertices inside int $\mathcal{C}$ ... but didn't we break ③?

↳ no because W is strongly $(5k-1)$-linked, so also $4k$-linked

→ so if previously int $\mathcal{C}$ contained $> \frac{2}{3}|W|$ vertices of W and now $\leq \frac{2}{3}|W|$, then S' contradicts W being $4k$-linked ... all components of $G - S'$ have size $\leq \frac{2}{3}|W|$

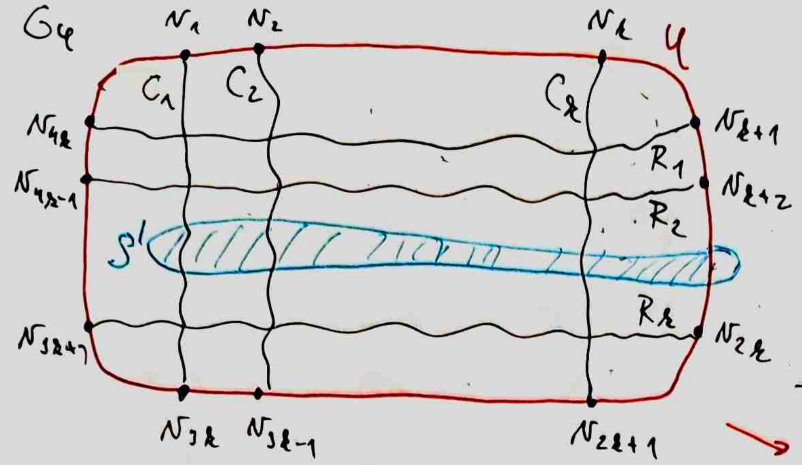
Denote the vertices along $\mathcal{C}$ by $v_1, v_2, \dots, v_{4k}$

Goal: Find k vertex-disjoint paths "columns" $C_1, \dots, C_k$ inside int $\mathcal{C}$, where

$C_1: v_1 \sim v_{3k}$, $C_2: v_2 \sim v_{3k-1}$, ..., $C_k: v_k \sim v_{2k+1}$

and similarly find k vertex disjoint "row" paths $R_1, \dots, R_k$ inside int $\mathcal{C}$, where

$R_1: v_{k+1} \sim R_{4k}$, $R_2: v_{k+2} \sim v_{4k-1}$, ..., $R_k: v_{2k} \sim v_{3k+1}$



$G_\mathcal{C}$ ... planar drawing of the subgraph of G with $\mathcal{C}$ as its outer face

↳ keep vertices on $\mathcal{C}$ & stuff inside

We will argue for the existence of the columns $C_1, \dots, C_k$

→ rows are the same argument

→ $S = \{N_1, N_2, \dots, N_{4k}\}$

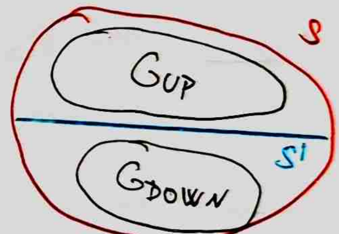
→ if the columns $C_1, \dots, C_k$ did not exist, by Menger's theorem there would be a <u>cutset S' of size $\leq k-1$</u> separating a source connected to $N_1, \dots, N_k$ from a sink connected to $N_{2k+1}, \dots, N_{3k}$

$S \cup S'$ is a good separator for W of size $\leq 5k-1$, so it is not strongly $(5k-1)$-linked ⚡

↳ $S \cup S'$ divides the graph inside int $\mathcal{C}$ into two components — G_{UP} and G_{DOWN}

- G_{UP} is separated from the rest of G by the cutset $S_{UP} := (S \setminus \{N_{2k+1}, \dots, N_{3k}\}) \cup S'$
- G_{DOWN} —— " —— $S_{DOWN} := (S \setminus \{N_1, \dots, N_k\}) \cup S'$

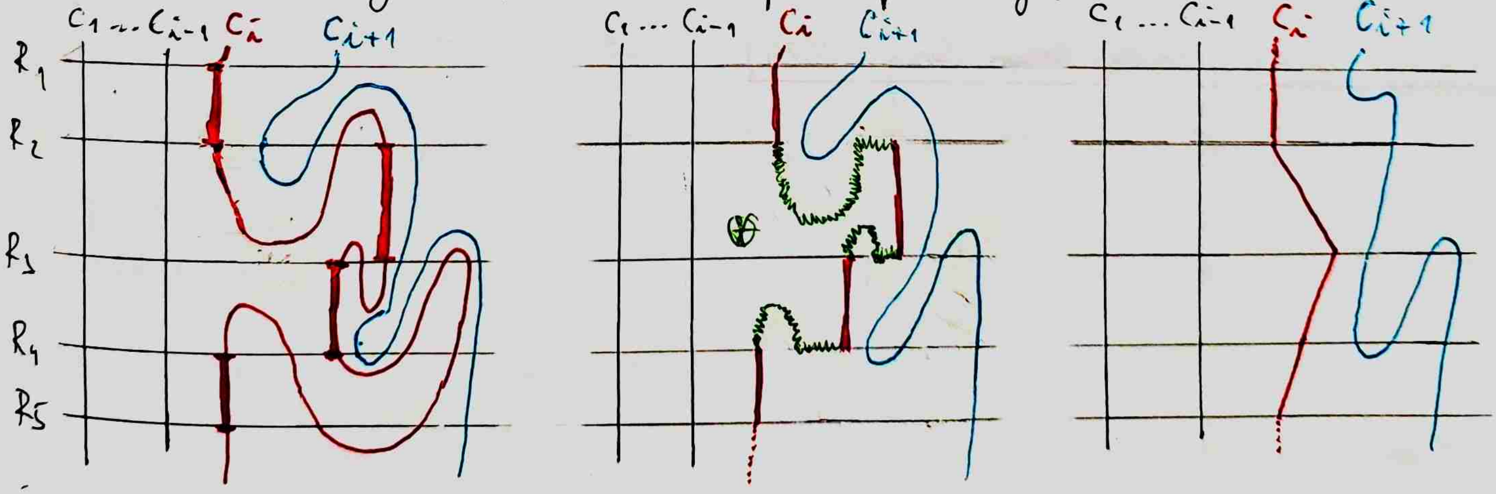
→ since $|S_{UP}|, |S_{DOWN}| \leq 4k$ and because $|G_{UP}|, |G_{DOWN}| <$ #vertices inside int $\mathcal{C}$, from minimality of S we must have $|G_{UP}|, |G_{DOWN}| \leq \frac{2}{3}|W|$



<u>Rearranging</u> $C_1, \dots, C_k$ and $R_1, \dots, R_k$ into a grid minor

→ first adjust the drawing so that $R_1, \dots, R_k$ are straight lines

→ we will inductively straighten $C_1, C_2, \dots$ from left to right



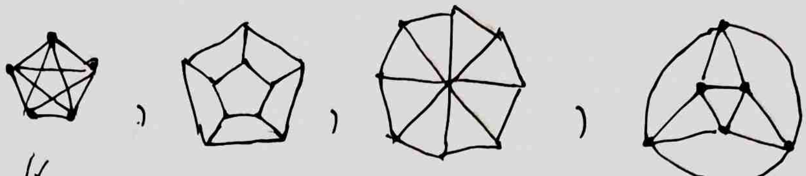
→ suppose we already have "straightened" $C_1, \dots, C_{i-1}$

→ for each pair of consecutive rows find the left-most segment of C_i connecting these rows

→ connect these segments along the boundary of the faces determined by the rows] ⊛

→ also adjust rows as needed so that these connections between the segments are parts of new rows. At the end we contract edges to identify ⊢↔⊣

TREE-WIDTH & FORBIDDEN MINORS

$tw(G) \leq 0$	K_2 	⇝ empty graphs
$tw(G) \leq 1$	K_3 	⇝ forrests
$tw(G) \leq 2$	K_4 	⇝ series-parallel graphs
$tw(G) \leq 3$		

K_5 , , , ↑ Wagner graph , $K_{2,2,2}$

HADWIGER CONJECTURE

Theorem (4-color theorem, 1976): $(K_5 \not\leq_m G \;\&\; K_3 \not\leq_m G) \Rightarrow \chi(G) < 5$

Conjecture (Hadwiger, 1943): $K_r \not\leq_m G \Rightarrow \chi(G) < 5$

- $r=2$: no edges $\Rightarrow \chi(G) = 1 < 2$
- $r=3$: no  $\Rightarrow$ forrests $\Rightarrow \chi(G) \leq 2 < 3$
- $r=4$: no  $\Rightarrow tw(G) \leq 2$ & $\chi(G) \leq tw(G)+1 \leq 3 < 4$
  - ↳ G is a partial 2-tree ... find 2-PES & color greedily
- $r=5$:  implies the 4-color theorem
  - ↳ Wagner (1937): the 4-color theorem implies $r=5$
- $r=6$: Robertson, Seymour, Thomas (1993) proved it from the 4-color theorem

Theorem (Delcourt, Postle, Norin, Song, 2024):

$$K_r \not\leq_m G \Rightarrow \chi(G) \in O(r \cdot \log\log r)$$

NICE TREE DECOMPOSITIONS

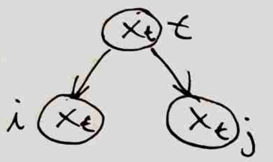
Def: A nice tree decomposition of G is a tree decomp (T, X) s.t.

① T is a rooted binary tree

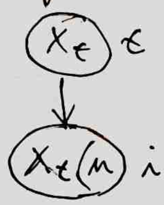
② every inner node $t \in T$ is one of:

- a join node with two children i, j s.t. $X_t = X_i = X_j$
- a forget node with one child i s.t. $X_i = X_t \cup \{u\}$, $u \notin X_t$
- an introduce node with one child i s.t. $X_t = X_i \cup \{u\}$, $u \notin X_i$

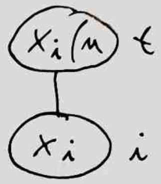
join node



forget node



introduce node

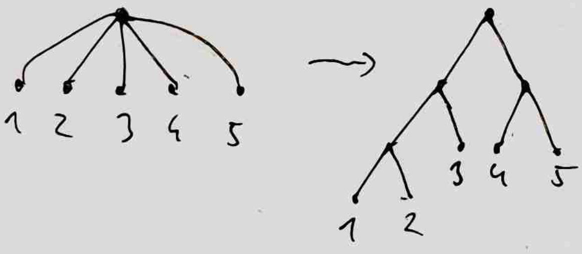


Remark: Sometimes it is required that if $t \in T$ is a leaf then $|X_t| = 1$

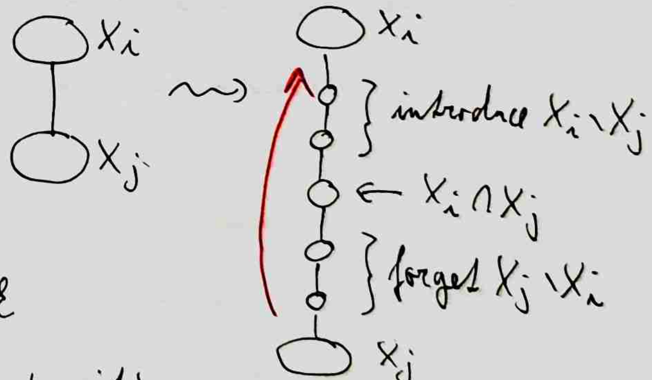
Observation: Any tree decomposition can be transformed in polynomial time into a nice tree decomp of the same width.

Proof: Root the tree arbitrarily ... (T, X) has width k

- if $\deg(t) > 2$, use $\deg(t) - 2$ join nodes



- replace each edge with $\leq k$ forget and $\leq k$ introduce nodes



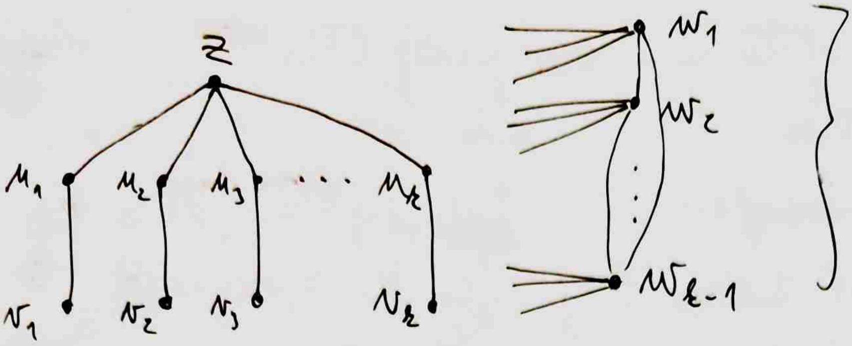
- if leaves should have size 1, replace each leaf with a path of introduce nodes of length $\leq k$

Corollary: Each graph G has a nice tree decomposition of optimal width on $O(|V_G| \cdot tw(G))$ nodes with all leaves size 1.

Proposition: $\forall k$ $\exists$ graph G on $3k$ vertices whose every optimal nice tree decomposition with leaves of size 1 has $\Omega(k^2)$ nodes.

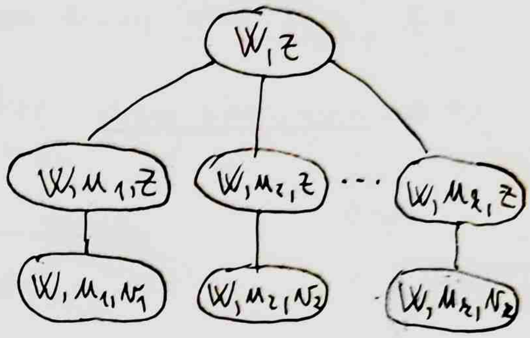
$\Rightarrow$ the above construction is asymptotically optimal ← WHEN LEAVES HAVE SIZE $= 1$

Proof: G looks like this: $W := \{w_1, \dots, w_{k-1}\}$



each w_i is connected to all other vertices

tree decomp of G of width = k:



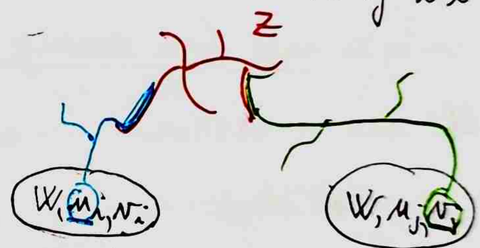
👁 $tw(G) = k$

↳ each set $W_i := \{u_i, v_i, w_1, \dots, w_{k-1}\}$ is a clique $\Rightarrow$ in $\forall$ tree decomp $\exists$ bag containing W_i $\Rightarrow$ $tw(G) \geq k$

Let (T, X) be a nice tree decomp of G of width k with leaves of size 1.

→ since each W_i is a clique, there $\exists\, t_i \in T$ s.t. $X_{t_i} = W_i$

→ since $z \notin W_i$, only 1 component of $T - t_i$ contains nodes with z in their bags and the bags of nodes in the other components are all subsets of W_i because $\forall j \neq i$: $T \upharpoonright u_j$ and $T \upharpoonright v_j$ intersect $T \upharpoonright z$ but $u_j, v_j \in W_i$ so these subtrees must be in the same component as $T \upharpoonright z$



👁 The $\subseteq$-minimal subtree of T containing nodes $t_1, \dots, t_k$ is still a valid (non-nice) tree decomp of G, where each t_i is a leaf

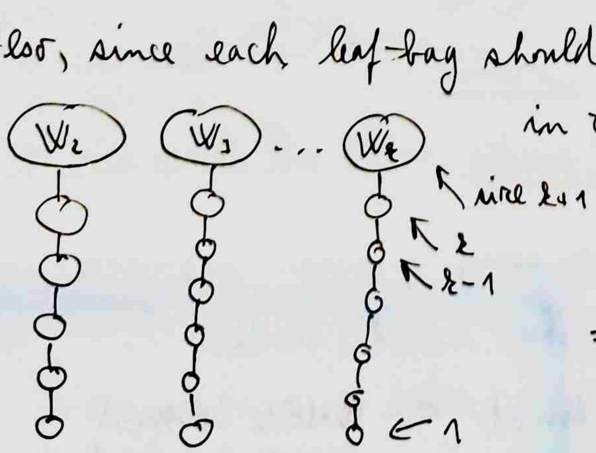
→ it looks something like this

👁 Does not matter how we root it, but at most one of $t_1, \dots, t_k$, say t_1, can be a predecessor (above) some other t_j

$\Rightarrow$ hence $t_2, \dots, t_k$ are in no predecessor-successor relation

Also, since each leaf-bag should have size = 1, each subtree of T rooted in t_i for $i = 2, \dots, k$ has $\geq k+1$ nodes below it

↳ namely at least k introduce nodes and at least 1 leaf node



$\Rightarrow |T| \geq (k-1)(k+2)$ ■

→ however, if we no longer require leaves of size 1, this counterexample breaks

Proposition: Every graph G has an optimal nice tree decomp. with $\leq 4 \cdot |V_G|$ nodes.

Proof: Assume that $tw(G) = k$, so G is a subgraph of an exact k-tree and there is $G' \supseteq G$ s.t. G' is a k-tree

$V_G = V_{G'} = V$

👁 a nice tree decomp. for G' is also nice for G →

→ let u be a simplicial vertex in G' whose neighborhood $N(u)$ induces a clique of size k

→ assume by induction that T' is a small, nice tree decomp. of $G-u$

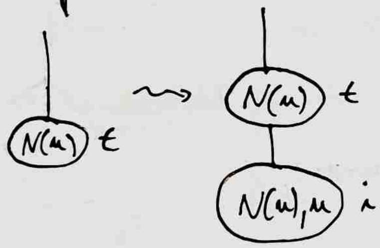
→ since $N(u)$ is a clique, there $\exists t \in T'$ s.t. $N(u) \subseteq X'_t$

We now distinguish 3 cases:

a) t is a join node: in that case, go into one of its two children (they have the same bags as t) and try again until we find a node that is not a join node

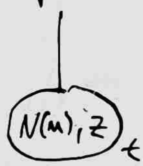
b) t is a leaf

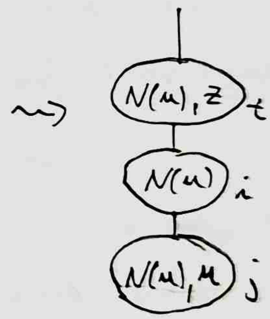
- if $X_t = N(u)$



added 1 node

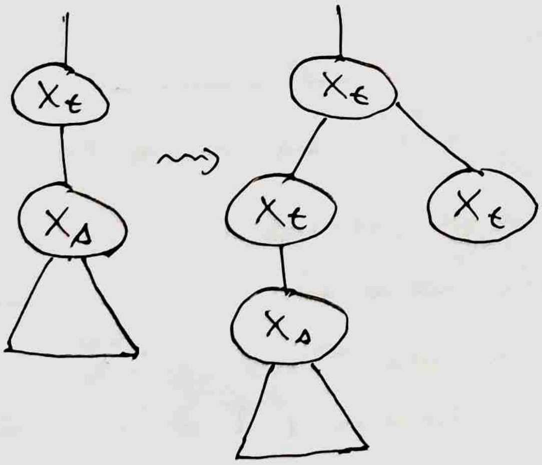
- if $X_t = N(u) \cup \{z\}$





added 2 nodes

c) t has 1 child



← this is now a leaf, so we can repeat case b)

⇒ in total we added 3 or 4 nodes

∎

PATHWIDTH

Def: A path decomposition of G is a tree decomp (T, X) where T is a path.

↳ path-width is defined analogously

$tw(G) \leq pw(G)$

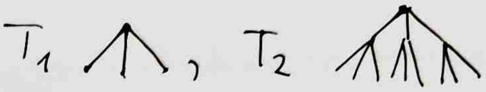
Proposition: Let T_k be the full ternary tree with $k+1$ levels.

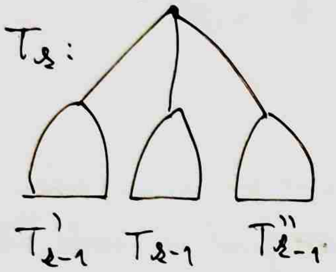
Then $pw(T_k) \geq k$.

bounded tw $\not\Rightarrow$ bounded pw

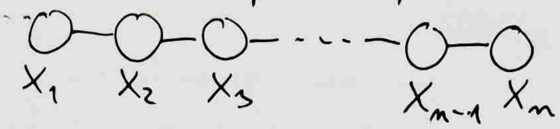
Proof: By induction. Case $k=1$ is easy

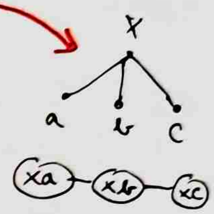
T_1, T_2





$\to$ let P be an optimal path decomp. of T_k



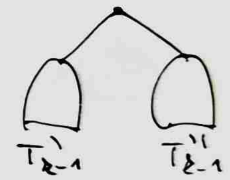


WLOG: X_1 contains some vertex of T'_{k-1}

X_n contains some vertex of T'_{k-1} or T''_{k-1}

Create a path decomp. P' of T_{k-1} by removing vertices of $T' := T \setminus T_{k-1}$ from bags in P. But in X_1 and X_n we have a vertex of T'.

Since T' is connected, every node in between ($\Rightarrow \forall$ node of P) contains at least 1 vertex of T'



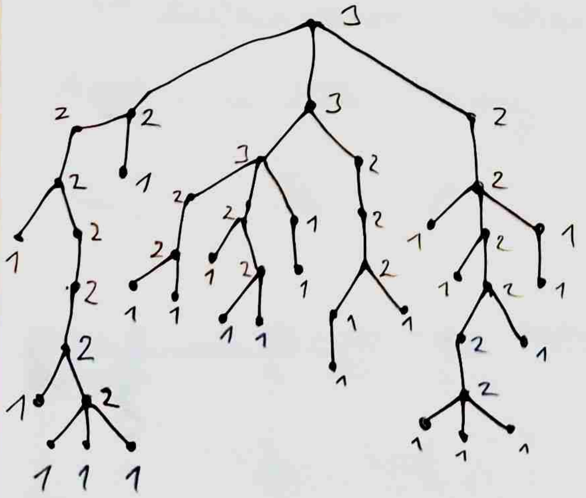
$\Rightarrow$ we deleted something from each bag $\Rightarrow pw(T_k) \geq pw(T_{k-1}) + 1$ ∎

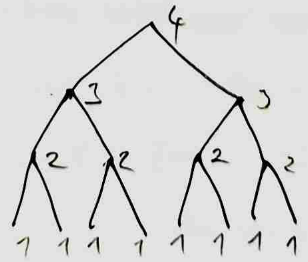
Theorem: If G is a graph on n vertices then $pw(G) \in O(tw(G) \cdot \ln(n))$.

Remark: In fact Θ because $pw(T_k) \geq k \approx \log_3 V(T_k)$ and $tw(T_k) = 1$

Proof: Let (T, X) be a tree decomp of G with $\leq n$ nodes.

↳ recall that such a decomposition exists (see section about k-trees)





$8+4+3+1 = 2^4 - 1$

- root T and define a labeling $\ell : T \to \mathbb{N}$
- t leaf $\Rightarrow \ell(t) := 1$
- t has 1 child $s \Rightarrow \ell(t) := \ell(s)$
- t has children $s_1, s_2, \dots$ with labels $\ell(s_1) \geq \ell(s_2) \geq \dots$

  $\ell(t) := \max(\ell(s_1), \ell(s_2) + 1)$

The smallest tree whose root has label n contains $2^n - 1$ nodes because a label is only increased when $\ell(s_1) = \ell(s_2)$

$\ell(r)$ … label of root $\rightsquigarrow$ $|V_T| \geq 2^{\ell(r)} - 1$ $\Rightarrow$ $|V_T| + 1 \geq 2^{\ell(r)}$

$\Rightarrow \ell(r) \leq \log_2(|V_T| + 1) \leq \log_2(n+1)$

We now use the labels to derive a path decomp (P, Y) of G

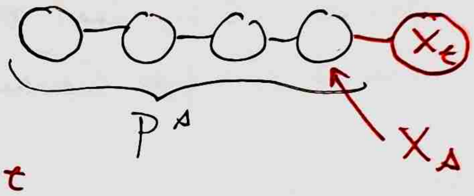
→ for $\forall t \in T$, consider the subtree T_t of T rooted in t

↳ $G_t :=$ subgraph of G induced by the vertices of $\bigcup\{X_i \mid i \in T_t\}$

↳ vertices in bags of T_t

→ recursively (going from leaves), we will make path-decomps (P^t, Y^t) of G_t … at the end $G = G_{ROOT}$

- t is a leaf: take a single node path decomp of G_t: P^t is X_t
- t has 1 child s: attach X_t to P^s:

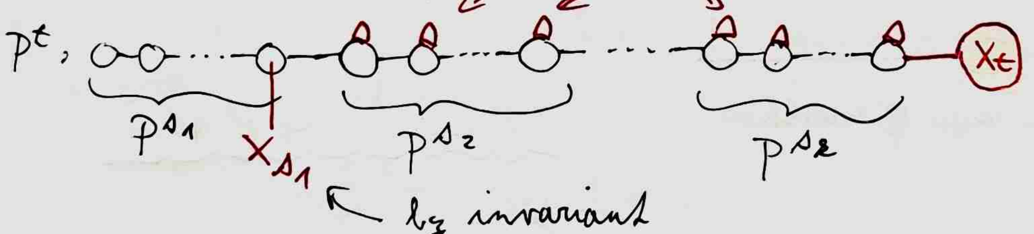


! construction invariant:

X_t is always a leaf bag of P^t

- t has children $s_1, \dots, s_k$ with $\ell(s_1) \geq \ell(s_2) \geq \dots \geq \ell(s_k)$

a) if $\ell(s_1) > \ell(s_2)$



b) if $\ell(s_1) = \ell(s_2)$: again concatenate $P^{s_1} \circ P^{s_2} \circ \dots \circ P^{s_k}$, add X_t as a leaf bag, but this time add vertices from X_t to $\forall$ bag of $P^{s_1}, \dots, P^{s_k}$

$\forall t$, the number of nodes of T whose bags were used to build bags of P^t is $\leq \ell(t)$

→ size of bag $\leq tw + 1$

$\Rightarrow$ this yields $pw(G) \leq \log_2(n+1)(tw(G)+1) - 1$

← -1 from definition of width

Remark: Graphs of pathwidth $\leq k$ are exactly those graphs where police with $k+1$ helicopters can always catch an invisible robber

↳ police moves without knowing the robbers position

↳ game ends when robber cannot move

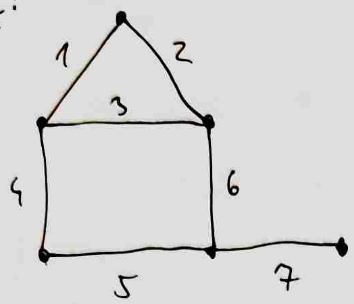
BRANCHWIDTH

$\nearrow |E| \geq 2$ $\qquad \nearrow$ inner nodes: $\deg = 3$

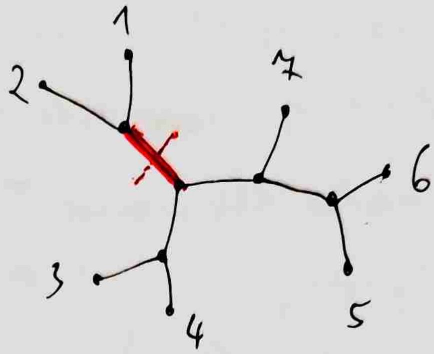
<u>Def</u>: A <u>branch decomposition</u> of $G = (V, E)$ is a <u>ternary tree</u> B with a bijection $\varphi : E \to$ leaves of B.

<u>Example</u>:

G



B



$\frac{21}{34567}$

every edge $f \in E_B$ defines a partition $E_G = E_f \,\dot\cup\, E'_f$

<u>Def</u>: The <u>width</u> of $f \in E_B$ is $\underline{w(f)} := |V_f| = |(\cup E_f) \cap (\cup E'_f)|$

$V_f :=$ vertices $v \in V_G$ that are an endpoint of both an edge from E_f and from E'_f

color E_f blue
E'_f red
$w(f) = \#$vertices

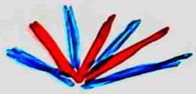
<u>Def</u>: The width of a branch decomp B is max width of $f \in E_B$

<u>Def</u>: The <u>branch-width of</u> G is $bw(G) :=$ min width of a branch decomp of G.

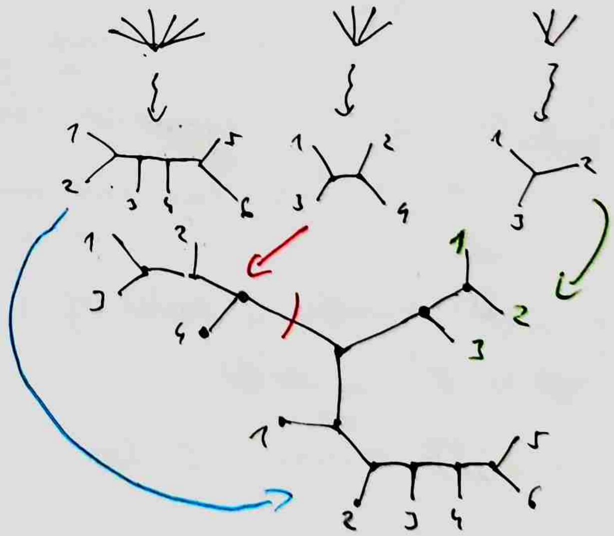
<u>Exercises</u>:

(1) <u>Which graphs have branch-width $= 1$</u>?

<u>stars</u> have all decomps of width $= 1$ since always $|V_f| \leq 1$

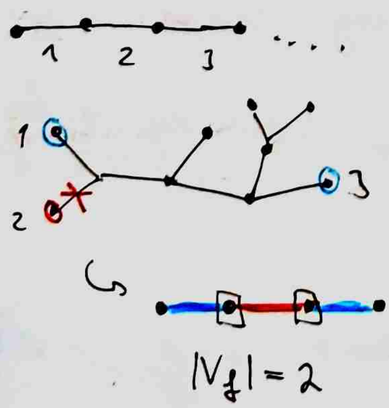


disjoint unions of stars

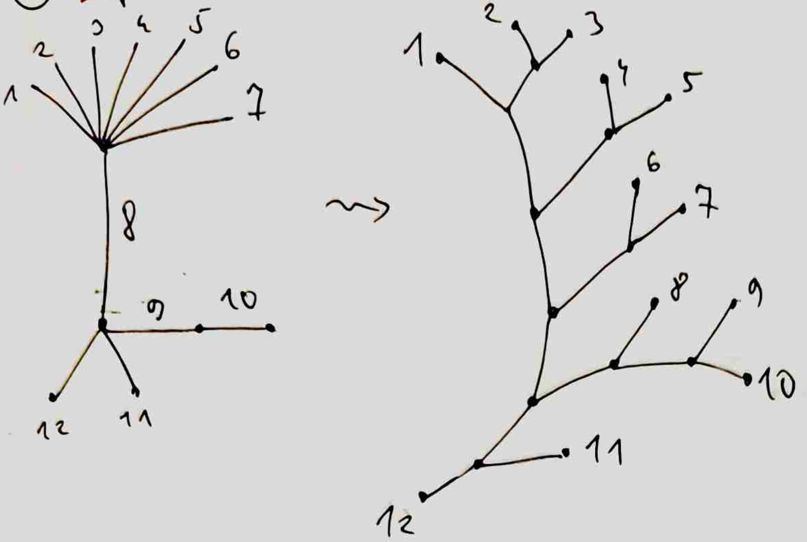


$\to$ all edges in the big one still have width $= 1$

If there $\exists$ <u>path of length 3</u>, branch-width ≥ 2



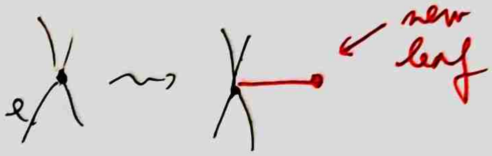
(2) If T is a tree then $bw(T) \leq 2$



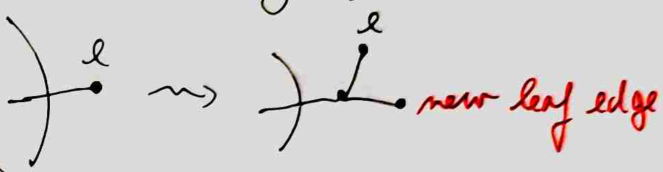
→ we construct T inductively by adding leaves

→ if $T =$ •—•—• (1, 2) use


→ if T was created from T' by adding a leaf to parent p



→ choose any edge incident with p

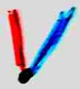


(3) $bw(K_n) \geq \frac{2}{3} n$ $n \geq 3$

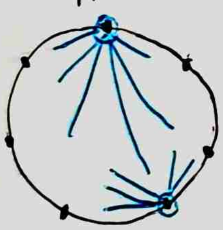
→ given a branch decomp B of G, choose the edge $f \in E_B$ that maximizes $\min(|E_f|, |E_f'|)$

↳ so # blue edges = # red edges (maybe up to a mistake of 1)

→ suppose it is the best branch decomp, so it minimizes # vertices



→ suppose we try to make as many only-blue vertices as possible → there cannot be an only-blue and an only-red vertex at the same time



$$\frac{n \cdot (n-1)}{2} = 1 + 2 + 3 + \cdots + \underline{(n-2) + (n-1) + n}$$

→ some tail $(n-m) + (n-m+1) + \cdots + (n-1) + n$ is blue edges

red = # blue ... for what k is $1 + 2 + \cdots + k = \frac{1}{2}(1 + 2 + \cdots + n)$?

$$\frac{n(n-1)}{2} = 2\,\frac{k(k-1)}{2} \Rightarrow k^2 \approx \frac{n^2}{2} \Rightarrow k \approx \frac{n}{\sqrt{2}}$$

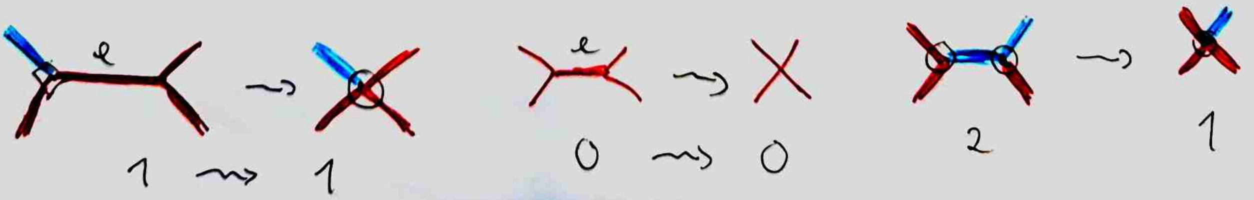
$\Rightarrow$ at least $\frac{n}{\sqrt{2}}$ are both red and blue, so $bw \geq \frac{n}{\sqrt{2}} = \frac{n}{1.41} \geq \frac{n}{1.5}$

(4) $H \leq_m G \Rightarrow bw(H) \leq bw(G)$

- delete edge ⇒ delete in B:

- delete vertex ⇒ delete incident edges
- contract edge ⇒ just delete the edge in B

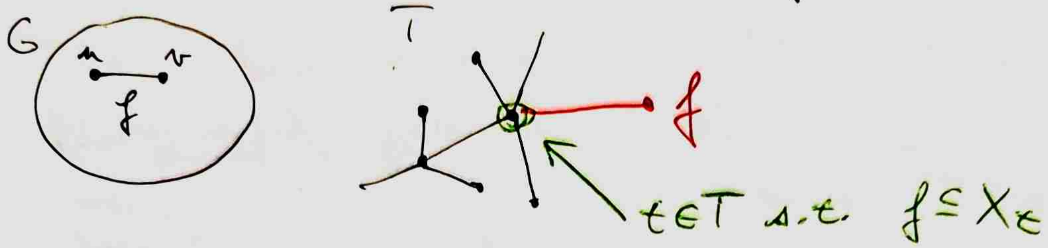
the width has not increased

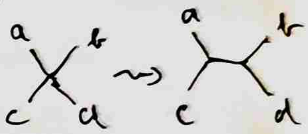


Theorem: If $bw(G) \geq 2$, then $\underline{bw(G) \leq tw(G)+1 \leq \frac{3}{2} bw(G)}$

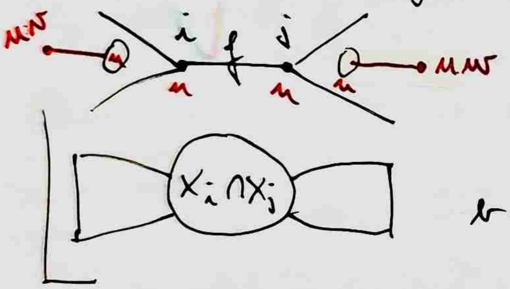
Proof: Let (T, X) be a tree decomp of G of width $= tw(G)$

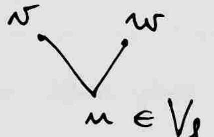
$\rightarrow$ we will construct a branch decomp of width $\leq tw(G)+1$ ↙ the -1 from def. of tree-width



- for $\forall f \in E_G$ find $t \in T$ s.t. $f \subseteq X_t$ and add a leaf representing f to t
- take the smallest subtree containing all the leaves that we added
  - ↳ in particular we delete all old leaves and keep only the leaves that represent an edge of G
- contract all vertices of degree 2 : 
- replace internal vertices of degree >3 by ternary subtrees : 

claim: The resulting tree B is a branch decomp of width $\leq tw(G)+1$



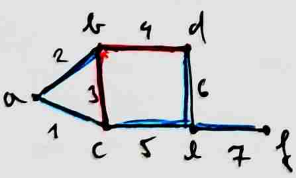


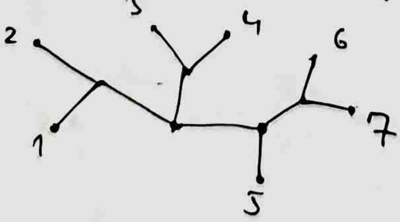
a) if f is a leaf edge of B then $|V_f| \leq 2$

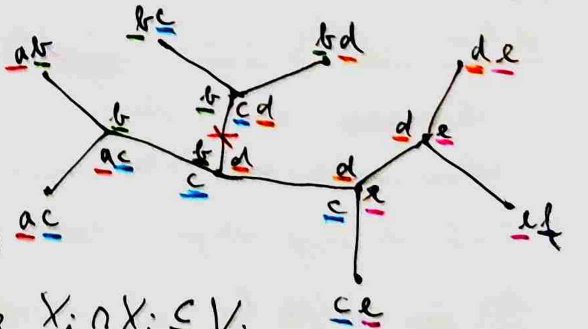
b) if $f \in E_B$ is connected to $i, j \in V_T$, then $V_f \subseteq X_i$ so $|V_f| \leq tw(G)+1$, since $X_i \cap X_j$ is a separator in G

To show the other inequality, let B be a branch decomp of G of width $= bw(G)$. We construct a tree decomp (T, X) where $T = B$ of G as follows:

- if a leaf $t \in B$ corresponds to the edge $uv \in E_G$, we let $X_t := \{u, v\}$
- if $uv, uw \in E_G$, insert u into every bag on the path from $\{u, v\}$ to $\{u, w\}$



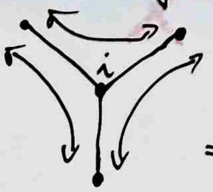




this is a tree decomp of G of width $\leq \frac{3}{2} bw(G)$

$\rightarrow$ clearly $ij \in E_T \Rightarrow bw(G) \geq |X_i \cap X_j|$ since $X_i \cap X_j \subseteq V_{ij}$

claim: if $i \in B$ is an inner node, then it has a neighbour j s.t. $|X_i| \leq \frac{3}{2} |X_i \cap X_j|$



its neighbors share parts of the communication going through i

$\Rightarrow$ choose the neighbour opposite least communication

↳ then $\frac{3}{2} bw(G) \geq |X_i|$ for $\forall i$ and we win ▨

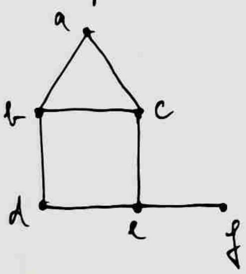
NLC-WIDTH & CLIQUE-WIDTH

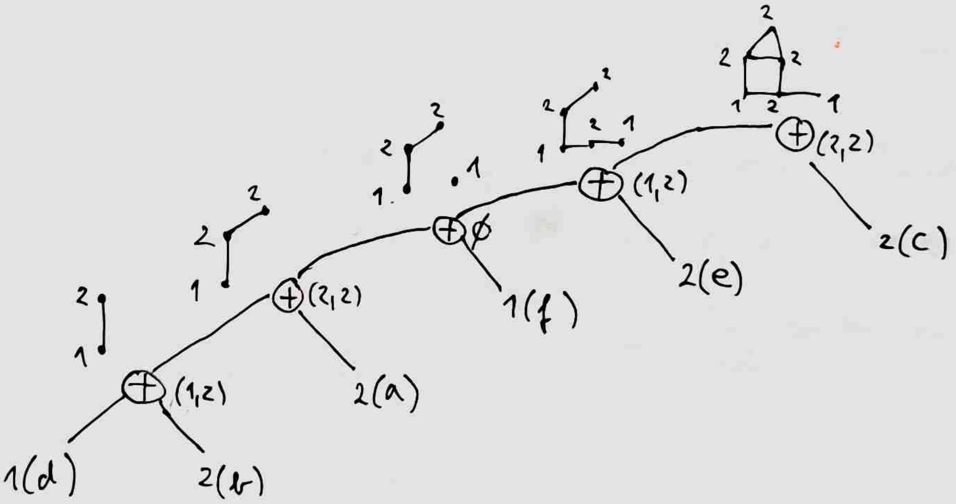
node-label controlled width

Def: Let $I = \{1, 2, \dots, k\}$ be a set of labels. An expression tree for a graph G is a binary rooted tree E (↳ all degrees ≤ 2) together with a mapping φ that assigns to each node $x \in E$ a labeled graph (G_x, ℓ_x) where $\ell_x : V(G_x) \to I$.
Moreover, the root of E has labeled graph (G_r, ℓ_r) where $G_r = G$.
The nodes of E are of three types i, ρ and $\oplus$:

- leaf node $i(v)$ corresponds to a single vertex graph $\overset{v}{\bullet}$ with label $\ell(v) = i$
- relabel node ρ_r where $r : I \to I$ has only 1 child (H, ℓ), and it corresponds to the graph (H, ℓ') where $\ell'(v) := r(\ell(v))$
  ↳ we chang each label j to $r(j)$
- join node $\oplus_S$ where $S \subseteq I \times I$ is a binary relation
  has 2 children (H, ℓ) and (H, ℓ'), and it corresponds to the graph (H^+, ℓ^+), where $\ell^+ = \ell \cup \ell'$ and to create H^+ first take the disjoint union $H + H'$ and then $\forall (u, v) \in V_H \times V_{H'}$ add edge if $(\ell(u), \ell'(v)) \in S$

Example:





Def: $nlcw(G) :=$ min. # labels necessary to construct an expression tree for G

Observations:

- $nlcw(G) \leq |V_G|$
- $nlcw(\overline{G}) = nlcw(G)$ ... instead of S take $I \times I \setminus S$ to build $\overline{G}$ instead (complement)
- $H \leq G \Rightarrow nlcw(H) \leq nlcw(G)$ ... delete unused leaf nodes and prune join nodes that now have only 1 child
  ↳ induced subgraph

Fact: nlcw is not minor monotone !
↳ $\forall n \; \exists H, G$ s.t. $nlcw(H) \geq n$, $nlcw(G) \leq 3$ & H is a minor of G

Def: Clique-width is defined similarly, but the expression tree is a bit different:

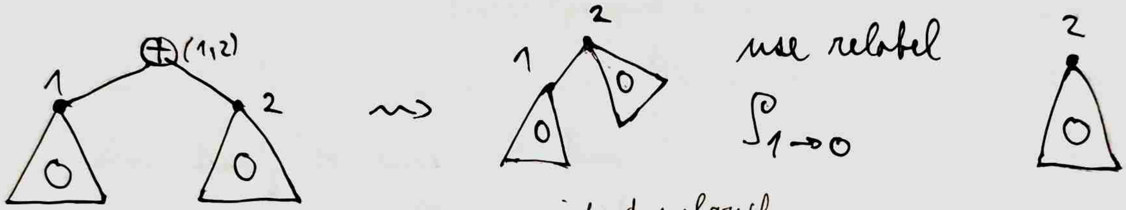
- leaf nodes $i(v)$ as before
- relabel nodes $\rho_{i\to j}$ only renames i to j, the rest of labels does not change
- $\oplus$ node only takes the disjoint union of the graphs of its two children
- $\eta_{i,j}$ node has 1 child H and adds edges between vertices $u, v \in V_H$ if $\ell(u) = i$ and $\ell(v) = j$

Exercise: $nlcw(G) \le cw(G) \le 2 \cdot nlcw(G)$

Examples:

① $nlcw(K_n) = 1$ ... just join everything or use $\overline{K_n} = E_n$ and $nlcw(E_n) = 1$

② $nlcw(\text{trees}) \le 3$ ... by joining trees & having a special label for the root



Lemma: $nlcw(G) = 1 \iff \forall H \le G$ (induced subgraph): if $|V_H| \ge 2$, then either H or $\overline{H}$ is disconnected

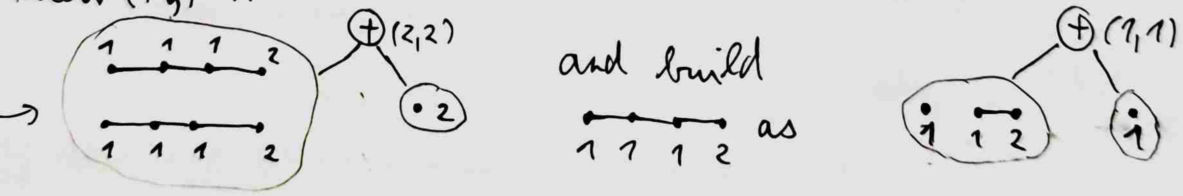
Fact: Graphs with this property are called cographs

↳ equivalently, G is a cograph $\iff P_4 \not\le G$

Clearly if G is a cograph then $P_4 \not\le G$ since $\overline{P_4} = P_4$ is not disconnected

Corollary: $n \ge 4 \Rightarrow nlcw(P_n) \ge 2$

Example: $nlcw(P_9) = 2$

↳ 

Proof of lemma:

$\Rightarrow$: look at the root of the subtree of the expression tree "induced" by H

↳ if $\oplus_\emptyset$, then H is disconnected; if $\oplus_{(1,1)}$ then $\overline{H}$ is disconnected

$\Leftarrow$ if $G = G_1 + G_2$ is disconnected then take $\oplus_\emptyset$ with children G_1, G_2 ← get expression trees recursively

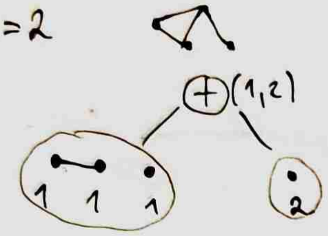
if G is connected then look at

$\overline{G} = H_1 + H_2$ and use $nlcw(G) = nlcw(\overline{G})$ ▨

Proposition: $tw(G) \le 2 \cdot nlcw(G) \cdot \Delta(G) - 1$ $\Delta(G)$ = max deg

→ take an expression tree and observe that whenever (G_x, l_x) is a graph corresponding to a node and $i \in I$ is a label s.t. $|\{v \in G_x \mid l_x(v) = i\}| > \Delta(G)$ then (G_x, l_x) is definitely not the child of a join node that would use the label i to create new edges

$\Delta = 2$



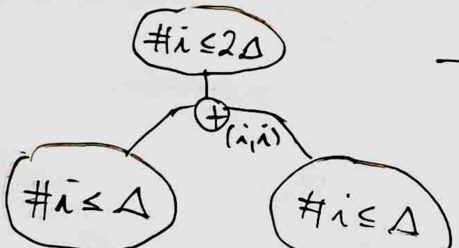
← impossible since we would create a vertex of degree $> \Delta(G)$

→ create a tree decomp (T, X) of G where T is the same underlying tree as used by the expression tree

→ define X_t by looking at the graph (G_t, l_t) and put

$$X_t := \{v \in G_t \mid \#l(v) \le 2\Delta\} \quad \text{where } \#i := |\{u \in G_t \mid l_t(u) = i\}| \text{ for } i \in I$$

↳ take those vertices whose label does not appear more than 2Δ times



→ we are adding edges only between labels that are used on $\le \Delta$ vertices

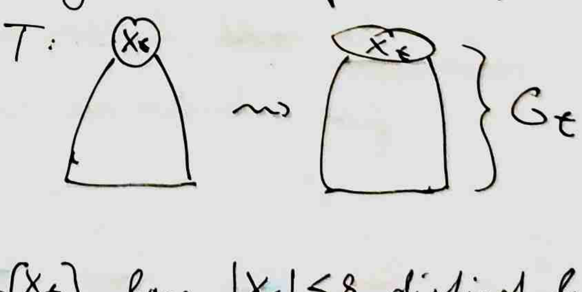
⇒ every vertex induces a connected subtree ▨

Theorem: $nlcw(G) \le 2^{tw(G)+1} + tw(G) + 1$ $k = tw(G) + 1$

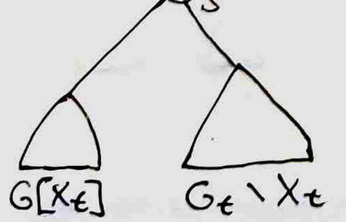
Proof: Let (T, X) be a nice tree decomp (with leaves of size $= 1$) of G of width $k-1$ with root r

→ recursively going from the leaves of T, we will construct an expression tree E for G, always satisfying the following invariant for $\forall t \in T$:

The expression tree constructed for $G_t :=$ subgraph of G induced by X_t and all its descendants



has the following form:



- vertices of $G[X_t]$ have $|X_t| \le k$ distinct labels
- vertices of $G_t \setminus X_t$ have $\le 2^k$ labels chosen so that

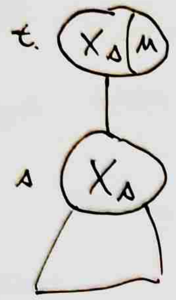
$$l(u) = l(v) \iff N(u) \cap X_t = N(v) \cap X_t$$

↳ they are adjacent to the same vertices from X_t

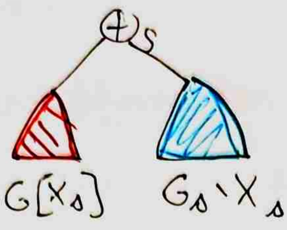
↳ we can now easily define S to add the correct edges

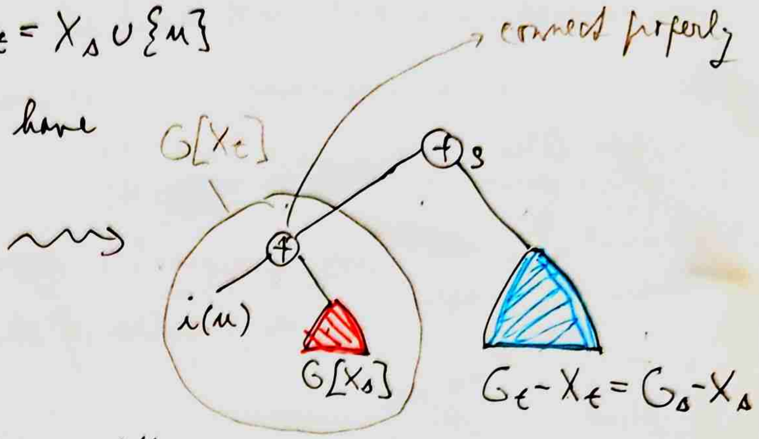
Constructing the expression trees: → T is a nice decomp.

- $t \in T$ is a leaf with bag $X_t = \{v\}$ ... take just $i(v)$
- $t \in T$ is an introduce node ... $X_t = X_s \cup \{u\}$



from induction we have

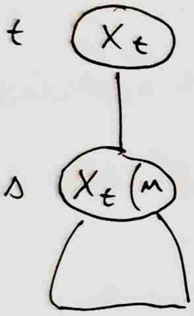




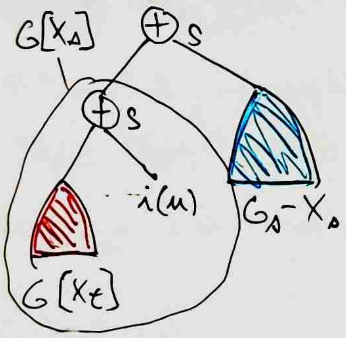
→ just add u to the left subtree with a new label & connect properly

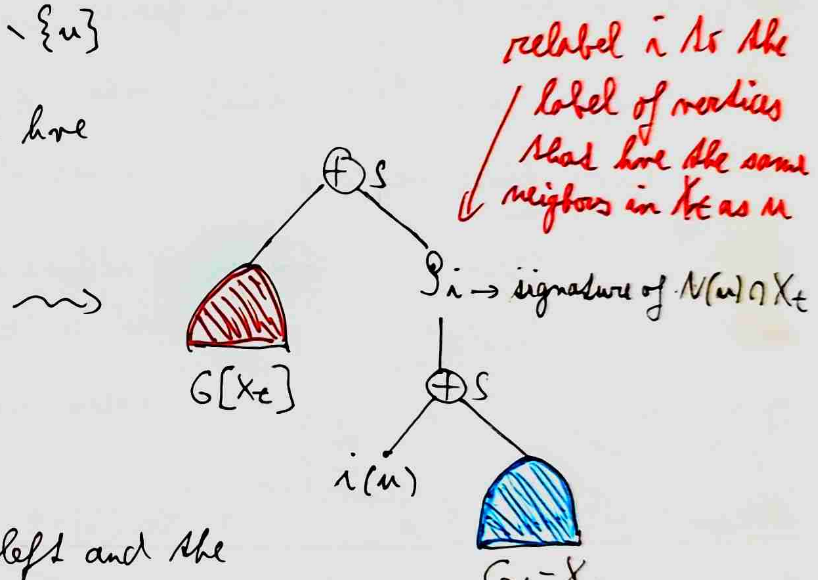
→ note that $\nexists v \in G_s \setminus X_s$ s.t. $uv \in E$ because X_s is a separator so we do not need to modify S

- $t \in T$ is a forget node ... $X_t = X_s \setminus \{u\}$



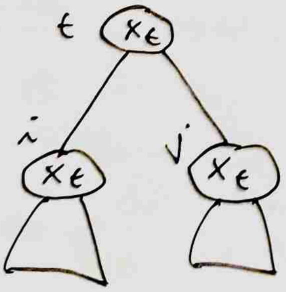
from induction we have



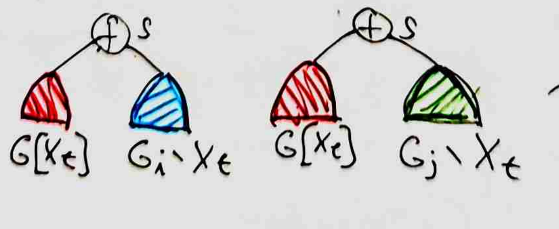


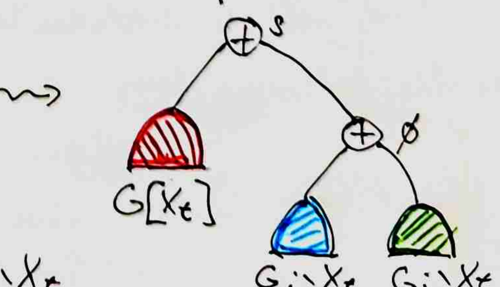
Note: Since the k labels used on the left and the 2^k labels used on the right are distinct, we can assume that the same relation S that is used on the root join node is also used on the join nodes in the left subtree

- $t \in T$ is a join node with children i and j, $X_t = X_i = X_j$



from induction we have expression trees for G_i and G_j





because X_t is a separator, $G_i \setminus X_t$ and $G_j \setminus X_t$ have disjoint vertex sets & there are no edges between them

⇒ we may WLOG assume that the two expression trees share the same relation S in their roots