# Parameterized algorithms

and

# Fine-grained complexity

[Part 2: Treewidth, planar graphs, logic.]

Michał Pilipczuk

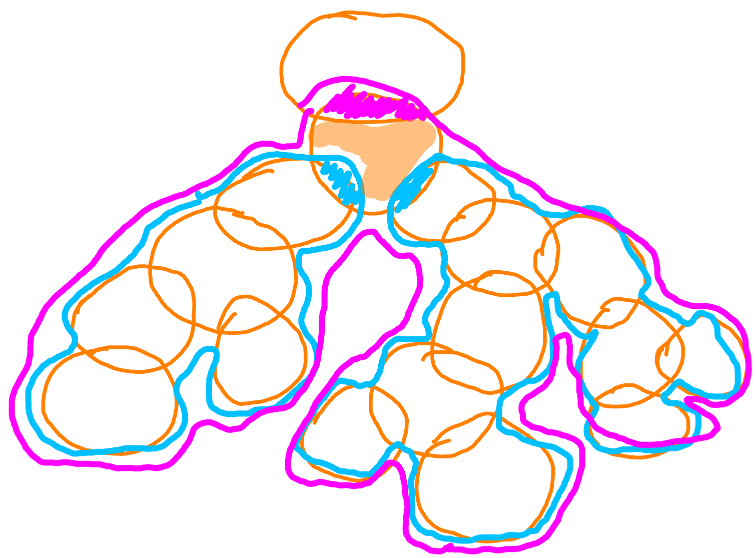
University of Warsaw

EPIT '23, Île d'Oléron

June 15th, 2023

# Structural parameters

Idea: decompose the graph into simpler pieces

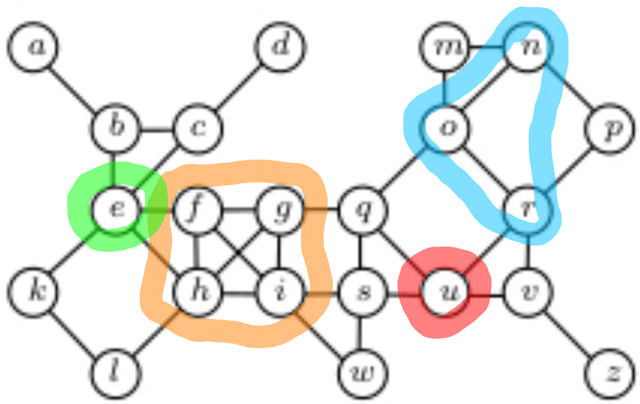


Need: - Every piece is simpler (parameter)
- Interaction between pieces is simple (parameter)

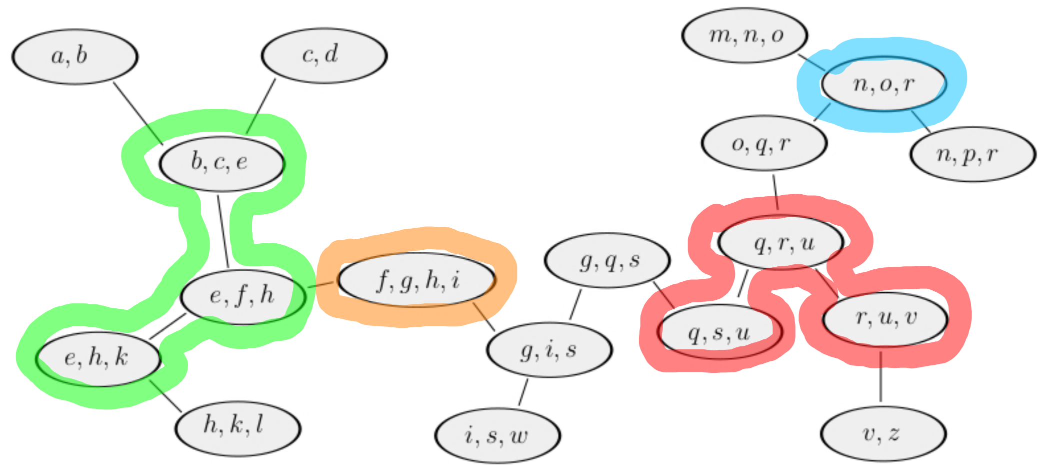
→ Process the decomposition bottom-up using DP

- Interesting on its own.
- Used as subroutines in larger algorithms.

# Tree decompositions




Graph $G$




Tree decomposition $(T, bag)$

**Tree decomposition:**

- tree $T$
- bag function $bag: V(T) \to$ subsets of $V(G)$

(T1) For every edge $uv$ of $G$, $u$ and $v$ appear together in a bag.

(T2) For every vertex $u$ of $G$, bags containing $u$ are connected.

<u>Intuition</u>: Smear vertices of $G$ to connected stains to reflect adjacency.

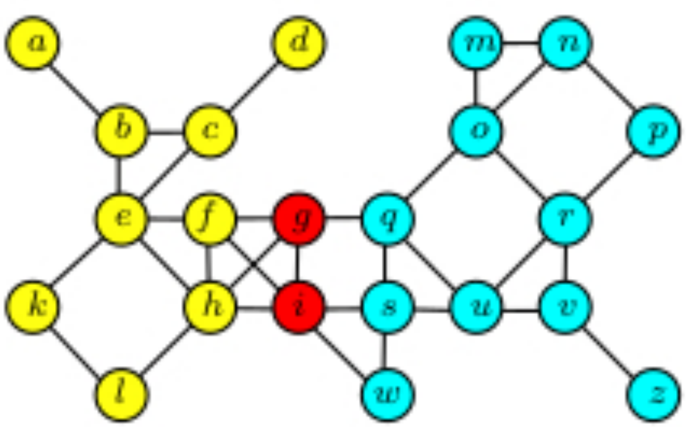
# Treewidth

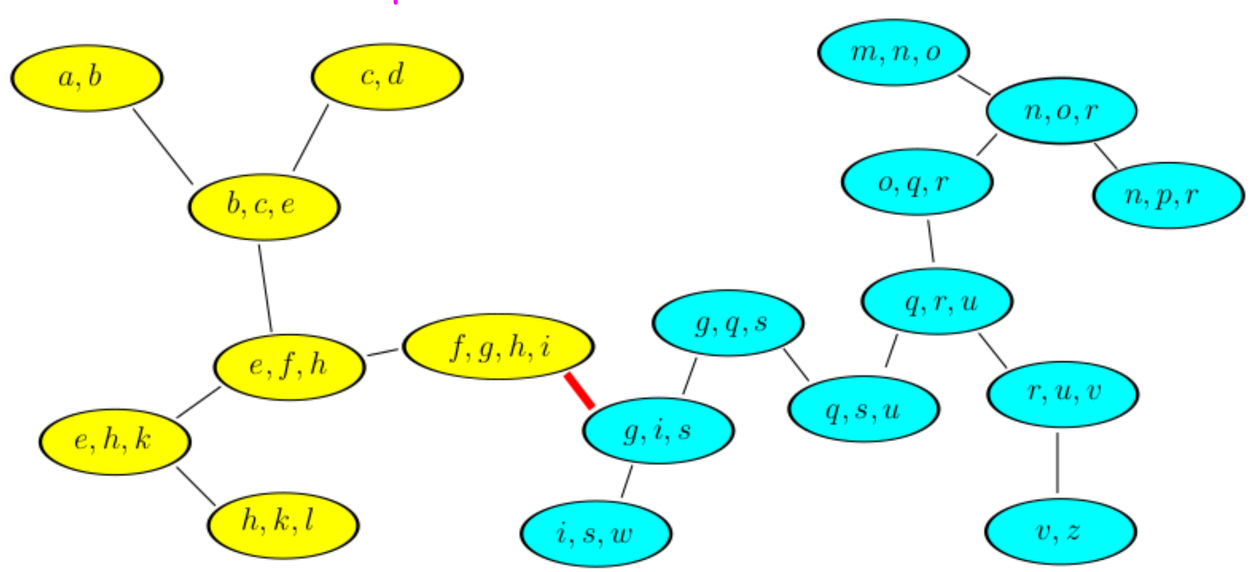
**Def** Width of $(T, bag)$ is max bag size $-1$.

**Def** Treewidth of $G$ is min width of a tree decomposition.

⌜ $G$ has treewidth $k$ if it can be squeezed into a shape of a tree of width $k+1$ ⌟

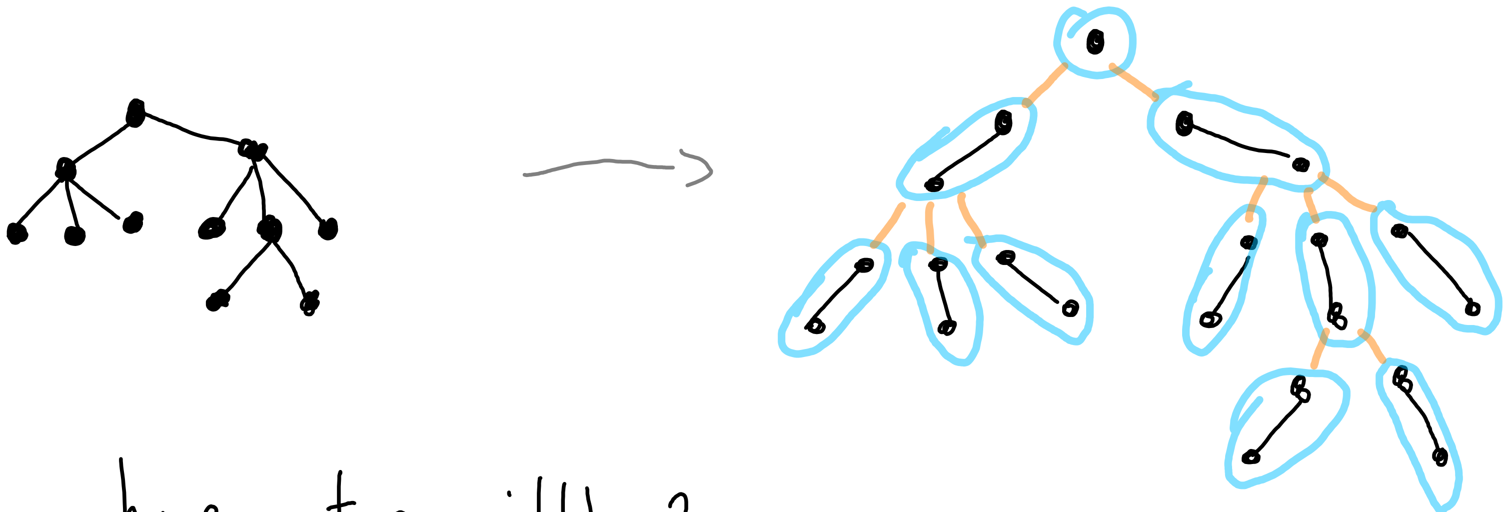
edges of $T$ $\longleftrightarrow$ separations in $G$



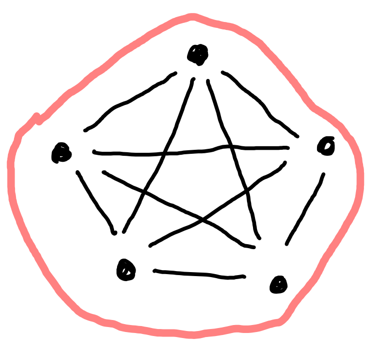
# Examples

(1) Trees have treewidth 1



(2) Cycles have treewidth 2

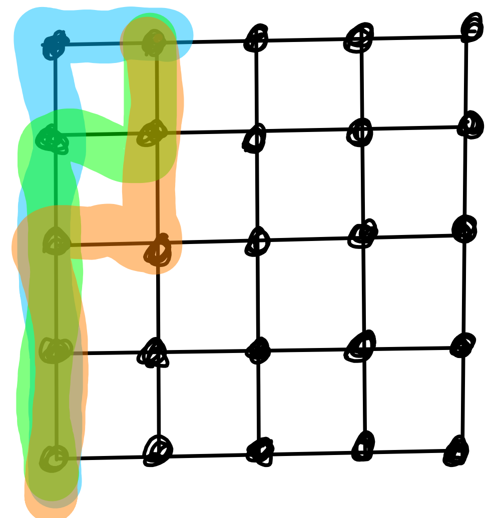
(3) Clique $K_t$ has treewidth $t-1$



Pf $t$ pairwise intersecting subtrees must have a bag in common $\square$

# Grids

(4) $t \times t$ grid has treewidth $t$



Sweep columns one by one

$\Downarrow$

$tw(Grid_{t \times t}) \leq t$

$tw(Grid_{t \times t}) \geq t$ $\rightsquigarrow$ Requires more arguments

**Cor** $G$ contains $Grid_{t \times t}$ as a subgraph

$\Rightarrow tw(G) \geq t$.

/* treewidth is a subgraph-monotone parameter */

# Minors

**Def** H is a minor of G if there is a minor embedding:

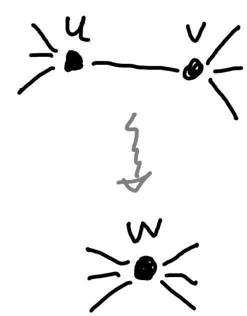

vertices $\longrightarrow$ disjoint connected subgraphs

edges $\longrightarrow$ edges connecting

**Fact** $H \preccurlyeq G \iff$ H can be obtained from G by
- vertex deletion
- edge deletion
- edge contraction

Pf) $\Rightarrow$ Contract branch sets, remove rest,

$\Leftarrow$ Track vertices under uncontractions. $\square$



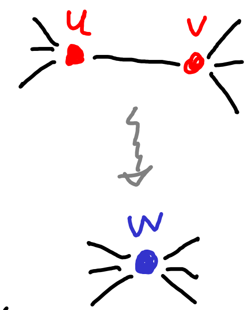

**Thm** [Kuratowski, Wagnor] G planar $\iff K_5, K_{3,3} \not\preccurlyeq G$.

# Minors and treewidth

**Fact** $H \preccurlyeq G \implies tw(H) \leq tw(G)$ /* tw is minor monotone */

*Pf* vertex deletion → trivial

edge deletion → trivial




edge contraction → replace $u$ and $v$ with $w$ in all bags $\square$

**Cor** $\mathrm{Grid}_{t \times t} \preccurlyeq G \implies tw(G) \geq t$

**Q** Is having a large grid minor <u>necessary</u> for having large treewidth?

# Grid Minor Theorem

**Thm** There is $f : \mathbb{N} \to \mathbb{N}$ such that

$$tw(G) > f(t) \implies \mathrm{Grid}_{t \times t} \preccurlyeq G.$$

Robertson and Seymour: No reasonable bound on $f$

Diestel, Jensen, Gorbunov, Thomassen: elegant $f(t) \leq 2^{O(t^5)}$.

Chekuri, Chuzhoy: $f(t) \leq \tilde{O}(t^{223})$

[[Current: Chuzhoy, Tan: $f(t) \leq \tilde{O}(t^9)$.]]

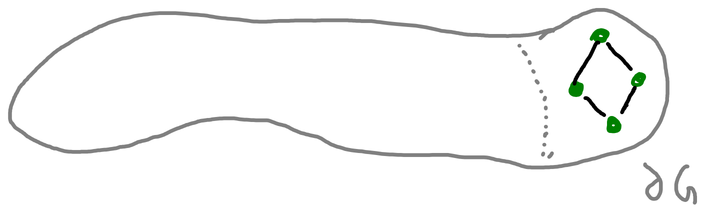
Lower bound: $f(t) \geq \Omega(t^2 \lg t)$

**Thm** If $G$ is planar, then $f(t) = 9/2\, t$ suffices.

⌜ Every graph is either tree-like or bidimensional ⌟

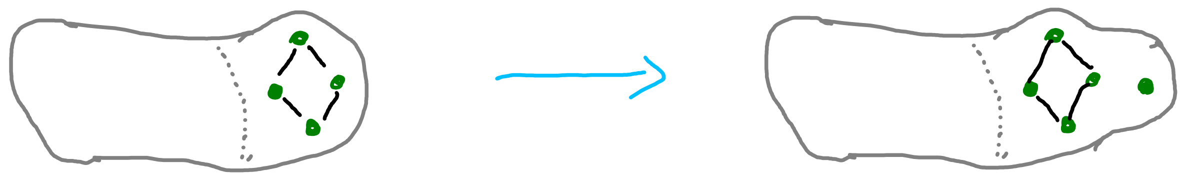
# Boundaried Graphs

**Def** $k$-boundaried graph = graph $G$ with boundary $\partial G \subseteq V(G)$ of size $\leq k$.

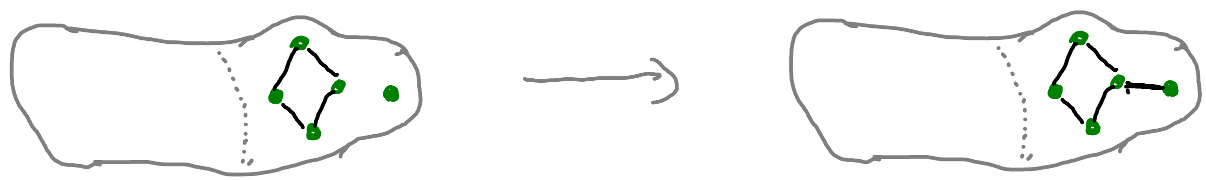



## Operations:

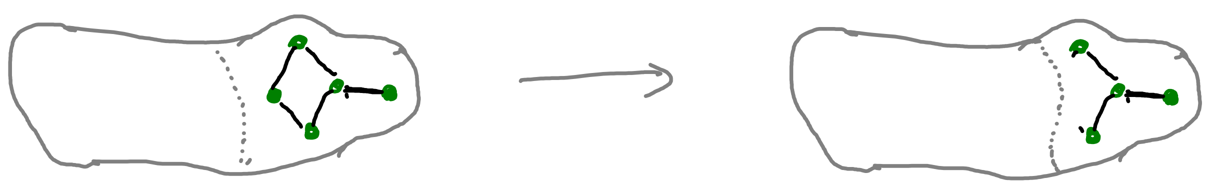
Introduce



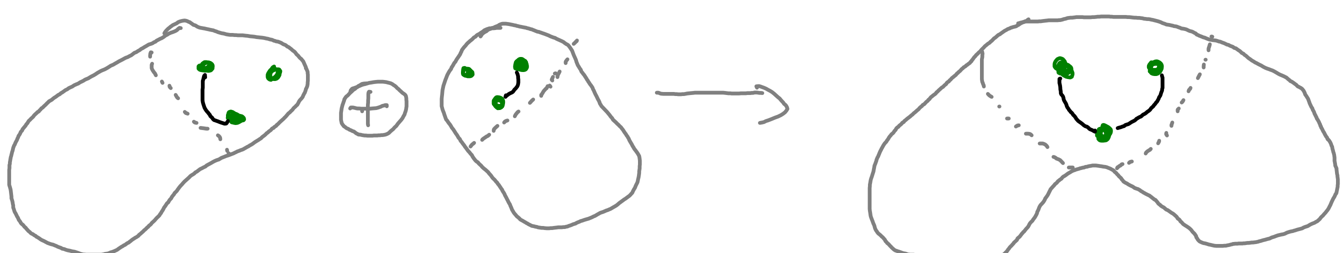
Add Edge



Forget



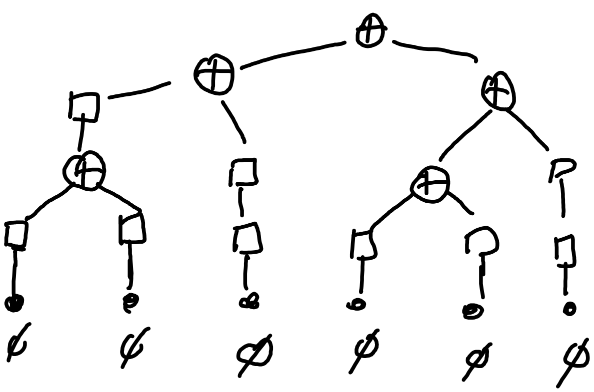
Join



# Boundaried graphs and tw

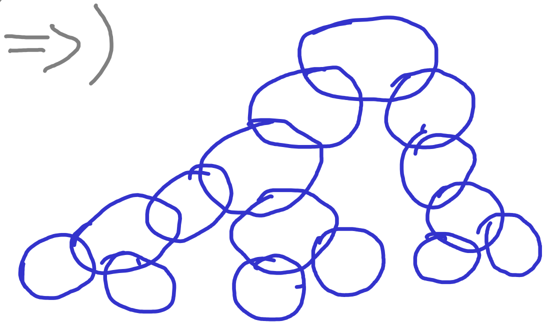
**Lemma** $tw(G) \leq t$ if and only if $G$ can be constructed from empty graphs in the algebra of $(t+1)$-boundaried graphs.

Pf ($\Leftarrow$) $\mathcal{T}$-term constructing $G$




- each node $x \to$ boundaried $G_x$
- $(\mathcal{T}, x \to \partial G_x)$ is a tree decomposition

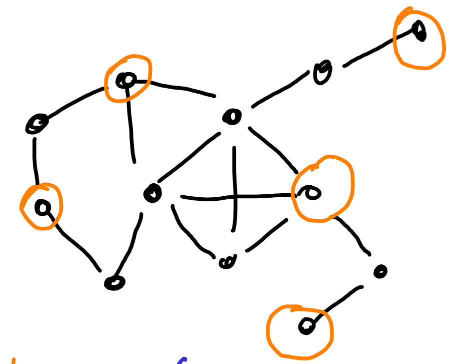
($\Rightarrow$)



- root decomposition $(T, bag)$
- construct larger and larger subtrees
- single step:
  - forget
  - introduce
  - join

□

# Dynamic Programming



## Max Independent Set

**Input:** graph $G$

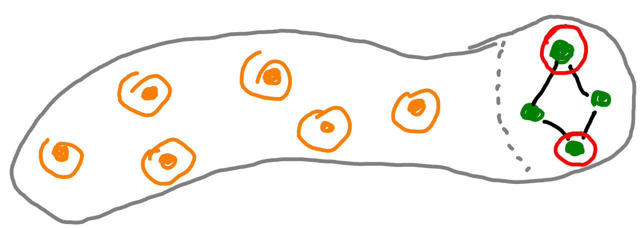
**Question:** maximum size of an independent set in $G$

**Thm** On a graph $G$ given together with a t.d of width $\le t$, Max Ind Set can be solved in time $2^t \cdot t^{O(1)} \cdot n$.

Pf $(T, bag) \rightsquigarrow$ term $\tau$ constructing $G$.

$$\Phi[G_x, \underset{\partial G_x}{A \cap\!\!|}] = \text{max size of } S \subseteq V(G_x) - \partial G_x \text{ s.t. } A \cup S$$

is independent



# Dynamic Programming

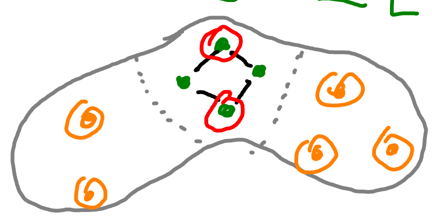
$$\Phi[G_x, \underset{\partial G_x}{\underset{\cap}{A}}] = \text{max size of } S \subseteq V(G_x) - \partial G_x \text{ s.t. } A \cup S \text{ is independent}$$

Need: $\Phi[G_{root}, \emptyset]$ assuming $\partial G_{root} = \emptyset$

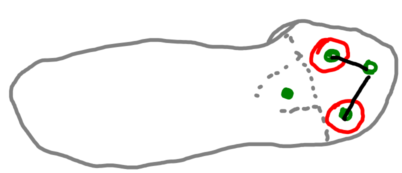
Have: $\Phi[\emptyset, \emptyset] = 0$

Recursion:

Join: $\Phi[G_1 \oplus G_2, A] = \Phi[G_1, A] + \Phi[G_2, A]$



Forget: $\Phi[G, A] = \max\left(\Phi[G^{old}, A],\ 1 + \Phi[G^{old}, A \cup \{\text{forgotten vrt}\}]\right)$



Introduce: ....

AddEdge: ....

# Dynamic Programming

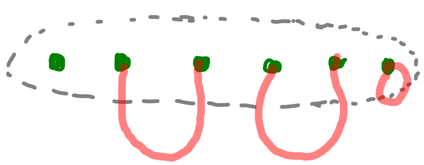
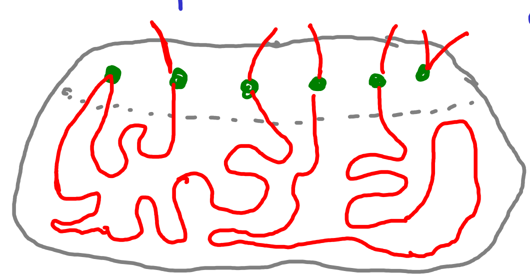
**Wrap-up:**

- $\mathcal{T}$ is of size $t^{O(1)} \cdot n$
- each node $\rightsquigarrow$ $\leq 2^t$ states to compute
- total time: $2^t \cdot t^{O(1)} \cdot n$ $\square$

⌜Principle: - Make a state for each possible behavior on boundary.
- Compute states in a bottom-up fashion.⌟

**Example:** - Hamiltonian Cycle

- behavior: set of disjoint pairs on $\partial G_x$
- $\Phi[G_x, M]$ = Are there paths realizing $M$ and visiting every vertex?

$\partial G_x$  $G_x$ 

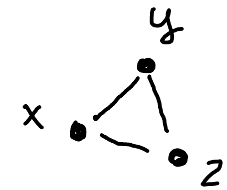
$\hookrightarrow t^{O(t)}$ states

# DPs for different problems

| | | Cut & Count |
|---|---|---|
| Ind Set | $2^t \cdot t^{O(1)} \cdot n$ | |
| Vertex Cover | $2^t \cdot t^{O(1)} \cdot n$ | |
| q-Coloring | $q^t \cdot t^{O(1)} \cdot n$ | |
| Dom Set | $3^t \cdot t^{O(1)} \cdot n$ | |
| FVS | $t^{O(t)} \cdot n$ $\rightsquigarrow$ | $3^t \cdot n^{O(1)}$ |
| Ham Cycle | $t^{O(t)} \cdot n$ $\rightsquigarrow$ | $4^t \cdot n^{O(1)}$ |
| k-Path | $t^{O(t)} \cdot n$ $\rightsquigarrow$ | $4^t \cdot n^{O(1)}$ |
| Subgraph Iso | $(t+3)^{\lvert V(H)\rvert} \cdot n$ | |

Courcelle's Theorem: A common explanation for $\uparrow$.

# Logic on Graphs




## First-Order (FO):

$$\text{triangleFree} := \neg \exists x \exists y \exists z \; \text{adj}(x,y) \wedge \text{adj}(y,z) \wedge \text{adj}(z,x)$$

## Monadic Second-Order, vertex variant ($MSO_1$)

$$\text{3colorable} := \exists_{X \subseteq V} \exists_{Y \subseteq V} \exists_{Z \subseteq V} \; \text{partition}(X,Y,Z) \wedge \text{ind}(X) \wedge \text{ind}(Y) \wedge \text{ind}(Z)$$

$$\text{partition}(X,Y,Z) := \forall u \; (u \in X \wedge u \notin Y \wedge u \notin Z) \vee \dots$$

$$\text{ind}(X) := \forall u \forall v \; \text{adj}(u,v) \Rightarrow (u \notin X \vee v \notin X).$$

## Monadic Second-Order, edge variant ($MSO_2$)

$$\text{HamCycle} := \exists_{C \subseteq E} \; \text{conn}(C) \wedge \forall_u \; \text{deg2}(u, C)$$

$$\text{deg2}(u,C) := \exists_e \exists_f \; e \in C \wedge f \in C \wedge \text{inc}(u,e) \wedge \text{inc}(u,f) \wedge e \neq f \wedge \dots$$

$\text{conn}(C) :=$ every nontrivial partition $(X,Y)$ of $V$ is crossed by $C$

# Courcelle's Theorem

$MSO_2$:
- variables for vertices, edges, sets of vertices, sets of edges
- atomic formulas: $u=v$, $u \in X$, $\mathrm{inc}(u,e)$, $\mathrm{adj}(u,v)$
- both universal and existential quantification

Sentence = formula without free variables

**Thm** [Courcelle]

Given sentence $\varphi \in MSO_2$ and $G$ with a t.d. of width $\le t$, one can check if $\varphi$ is true in $G$ in time $f(\varphi, t) \cdot n$.

computable

Sketch: $(T, bag) \rightsquigarrow$ term $\mathcal{J} \rightsquigarrow \Sigma_t$-labelled tree $S$

$\varphi \rightsquigarrow \varphi'$ working on $S \rightsquigarrow$ tree automaton $\mathcal{A}_{\varphi'}$

Then just run $\mathcal{A}_{\varphi'}$ on $S$. $\square$

Connection of PA with automata!

Note: Can also optimize $|A|$ subject to $G \models \varphi(A)$.

# Computing tree decompositions

To use a tree decomposition, one needs to compute one first.

Bad news: - NP-hard to compute exactly
- No constant factor apx under small set expansion conj.

Good news: - Variety of fpt exact and apx algorithms

## Selection

| authors | runtime | output width |
| --- | --- | --- |
| Robertson & Seymour, 80s | $O(8^t \cdot t^2 \cdot n^2)$ | $4k+3$ |
| Bodlaender '95 | $2^{O(k^3)} \cdot n$ | $k$ (exact) |
| Korhonen '21 | $2^{O(k)} \cdot n$ | $2k+1$ |

# Piecing things together

We prove that Feedback Vertex Set is FPT.

Input: Graph $G$, integer $k$

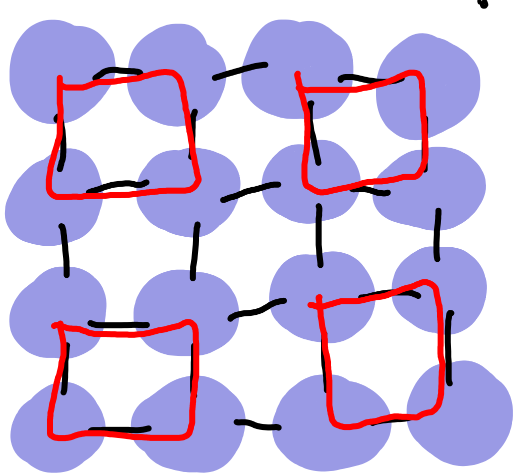
grid minor function

**Step 1** Run tw approximation for $t = f(2\sqrt{k+1})$.

→ Case 1: tree decomposition of width $\leq 2t+1$

→ Run DP in time $t^{O(t)} \cdot n = 2^{\tilde{O}(k^{4.5})} \cdot n$

→ Case 2: $tw(G) > f(2\sqrt{k+1}) \Rightarrow 2\sqrt{k+1} \times 2\sqrt{k+1}$ grid minor

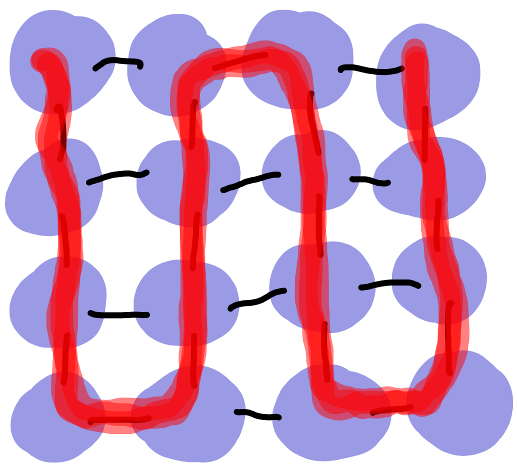


→ Answer NO

because each of $\sqrt{k+1} \cdot \sqrt{k+1}$ cycles must be hit.

# Bidimensionality

Same approach works for k-path:



$\sqrt{k} \times \sqrt{k}$ grid minor $\Rightarrow$ YES

More difficult applications:

- grid does not immediately give an answer
- argue that middle vertex is irrelevant

Example: Planarization (delete k vertices to a planar) is FPT.

---

Now assume that G is planar. $\rightsquigarrow$ $f(t) \le O(t)$

k-path runtime: $f(\sqrt{k})^{O(f(\sqrt{k}))} \cdot n = k^{O(\sqrt{k})} \cdot n \overset{C\&C}{\rightsquigarrow} 2^{O(\sqrt{k})} \cdot n$

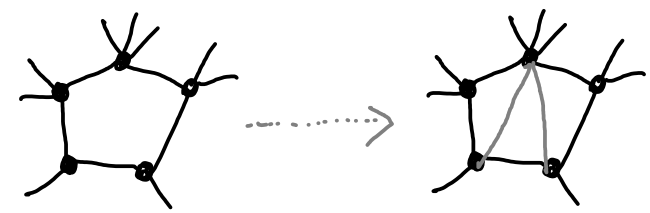
FVS runtime: $k^{O(\sqrt{k})} \cdot n \overset{C\&C}{\rightsquigarrow} 2^{O(\sqrt{k})} \cdot n$

[[ Bidimensionality $\leftrightsquigarrow$ $\sqrt{\ }$ Phenomenon ]]

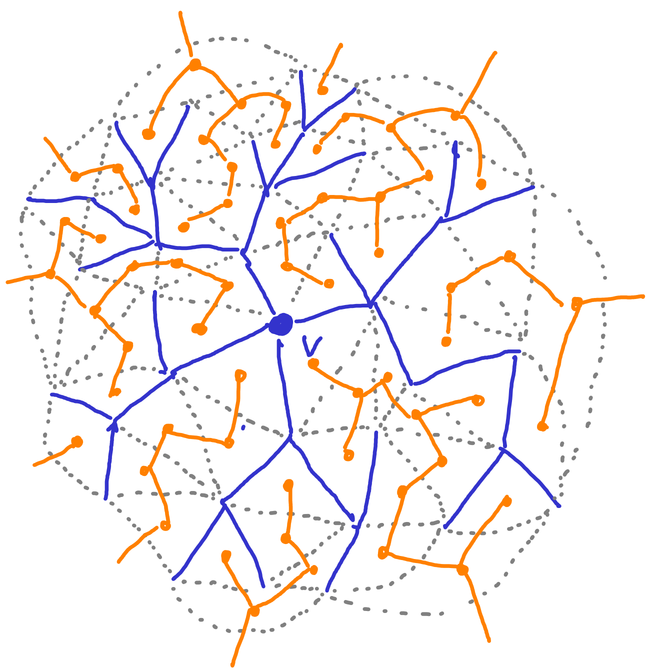
# Treewidth and Radius

**Thm** If $G$ is a connected planar graph of radius $\leq d$,
then $tw(G) \leq 3d$.

Pf Wlog $G$ is triangulated



Run BFS from the center $v$ ⇝ Tree $T$ of depth $\leq d$.



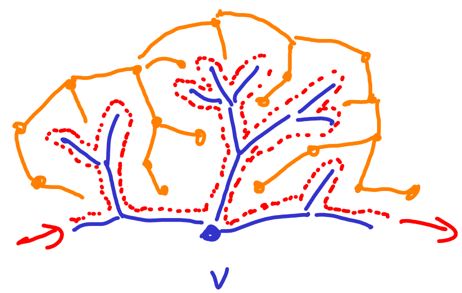

Look at edges of $E(G) - E(T)$

$S :=$ graph on Faces$(G)$ with
edges dual to edges of $E(G) - E(T)$

**Claim:** $S$ is also a tree

# Treewidth and Radius

**Claim:** $S$ is also a tree.

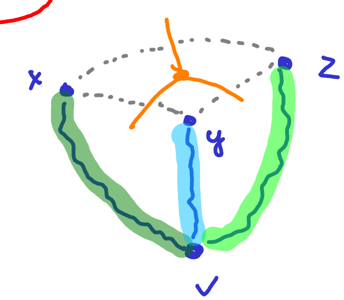



**Obs 1** $S$ is connected.

**Obs 2** #edges of $S$ is

$$|E(G)| - |E(T)| = |E(G)| - |V(G)| + 1 \overset{\text{Euler}}{=} |F(G)| - 1 = |V(S)| - 1 \qquad \square$$

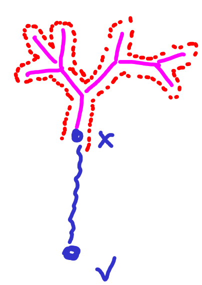
**Tree decomposition:** $\left(S,\ xyz \longrightarrow \underbrace{V(P_x) \cup V(P_y) \cup V(P_z)}_{\text{size} \leq 3d+1}\right)$




(T1) Every edge covered $\longrightarrow$ Trivial

(T2) Every blob connected

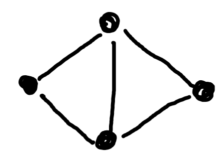
↳ Partial Euler tour



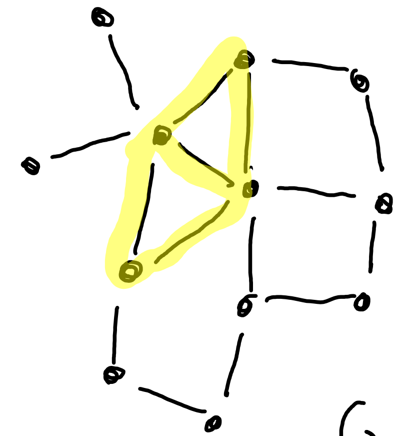
# Subgraph Isomorphism

**Input:** Graphs $H$ and $G$, $|V(H)| = k$

**Question:** Is $H$ a subgraph of $G$?



H



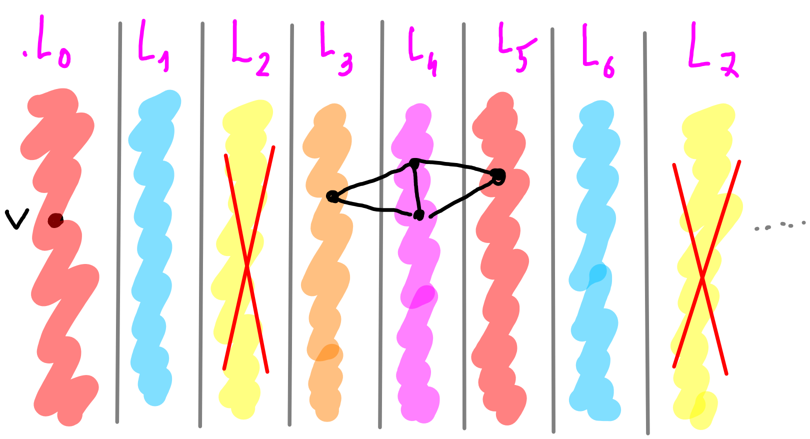
G

**Thm** $SI/k$ is FPT on planar graphs.

layering,
shifting,
Baker's technique

Pf /* Assumption: $G$ is connected */

Run BFS from any $v$, divide into Layers




Color layers mod $k+1$

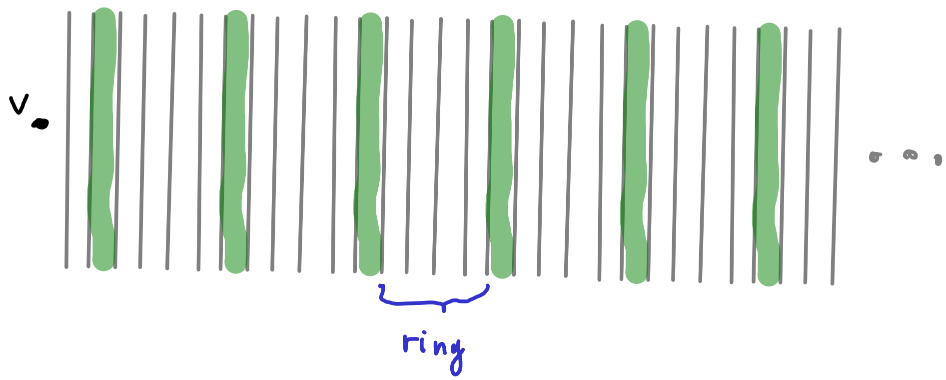
**Obs** $\exists$ solution .

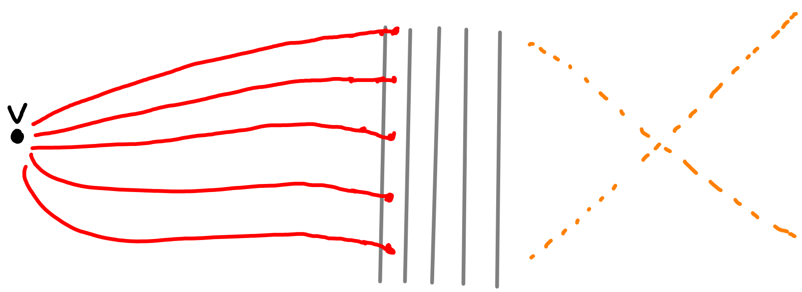
$\Rightarrow \exists$ color not intersecting it

$\hookrightarrow$ Guess it and remove it!

# Subgraph Isomorphism

**Claim** After removing any color, $tw \leq 3k$.




**Obs** Suffices to focus on a single ring.

- contract earlier layers on $v$
- remove later layers

$\to$ **Planar** graph of radius $\leq k$

$\Downarrow$

$tw(\text{ring}) \leq 3k \Rightarrow tw(G - \text{color}) \leq 3k$ $\square$

**Now** Apply DP in time

$$O\left((|V(H)|+3)^{tw} \cdot n\right) = k^{O(k)} \cdot n$$

$\square$

# SI ⇝ Logic

## Subgraph Isomorphism:

$$\exists x_1 \exists x_2 \exists x_3 \ldots \exists x_k \; adj(x_1, x_2) \wedge adj(x_2, x_3) \wedge \ldots\ldots$$

↳ Checking an existential FO sentence

## FO Model Checking

Input: Graph $G$, sentence $\varphi \in FO$

Question: Is $\varphi$ true in $G$?

SI is fpt on bnd degree ⟶ FO MC is fpt on bnd degree

SI is fpt on planar ⟶ FO MC is fpt on planar

⟶ bnd expansion ⟶ nowhere dense ⟶ ⋯

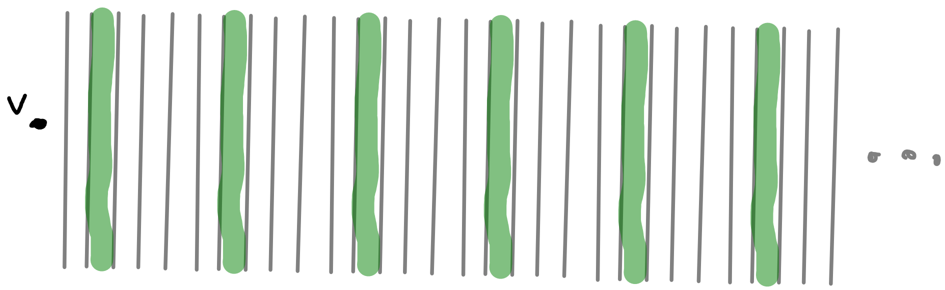
⌜Courcelle: $MSO_2$ model checking is fpt on classes of bnd treewidth⌟

# Approximation schemes

**Given:** planar $G$, $\varepsilon > 0$

**Goal:** find independent set of size $\geq (1-\varepsilon)\,OPT$

wlog assume $G$ is connected, apply layering from any $v$




Color layers mod $1/\varepsilon$

$\Downarrow$

There is a color that contains $\leq \varepsilon$ fraction of OPT

$\Leftarrow$ Removal of this color leaves a graph with solution of size $\geq (1-\varepsilon)\,OPT$ and $tw \leq 3/\varepsilon$

**Ergo:** Guess the color, remove, find optimum in time $\underbrace{2^{O(1/\varepsilon)} \cdot n}_{\text{total time complexity}}$.

EPTAS $\rightsquigarrow$ Running time $f(\varepsilon) \cdot n^{O(1)}$ $\rightsquigarrow$ FPT

PTAS $\rightsquigarrow$ Running time $f(\varepsilon) \cdot n^{f(\varepsilon)}$ $\rightsquigarrow$ XP