# Uniform Kernelization Complexity of Hitting Forbidden Minors

Archontia C. Giannopoulou

University of Warsaw, Poland

Algorithmic Graph Theory
on the
Adriatic Coast

June 18, 2015

Joint work with Bart Jansen, Daniel Lokshtanov, and Saket Saurabh

# F-Minor-Free Deletion

**F-Minor-Free Deletion**

| | |
|---|---|
| Input: | A graph $G$ and a positive integer $k$. |
| Parameter: | $k$. |
| Question: | Does there exist a set $S \subseteq V(G)$ such that $\lvert S \rvert \leq k$ and $G \setminus S$ doesn't contain any of the graphs in $\mathcal{F}$ as a minor? |

# F-Minor-Free Deletion

## F-Minor-Free Deletion

| | |
|---|---|
| Input: | A graph $G$ and a positive integer $k$. |
| Parameter: | $k$. |
| Question: | Does there exist a set $S \subseteq V(G)$ such that $\lvert S \rvert \leq k$ and $G \setminus S$ doesn't contain any of the graphs in $\mathcal{F}$ as a minor? |

Examples:

- Vertex Cover
- Feedback Vertex Set
- Vertex Planarization
- Treewidth-$\eta$ Deletion
- Treedepth-$\eta$ Deletion

# F-MINOR-FREE-DELETION

- $\mathcal{F}$-MINOR-FREE DELETION is FPT.
  [Robertson and Seymour, Journal of Combinatorial Theory B]

# F-Minor-Free-Deletion

- $\mathcal{F}$-Minor-Free Deletion is FPT.
  [Robertson and Seymour, Journal of Combinatorial Theory B]

If $\mathcal{F}$ contains a **planar** graph (Planar F-Minor-Free-Deletion)

| | |
|---|---|
| Constant factor approximation algorithm | [Fomin et al, FOCS 2012] |
| $2^{\mathcal{O}(k \log k)} n$ algorithm | [Fomin et al, FOCS 2012] |
| $f(\mathcal{F}) k^{O(g(\mathcal{F}))}$ kernel | [Fomin et al, FOCS 2012] |
| | |
| | |

# F-Minor-Free-Deletion

- $\mathcal{F}$-Minor-Free Deletion is FPT.
  [Robertson and Seymour, Journal of Combinatorial Theory B]

If $\mathcal{F}$ contains a **planar** graph (Planar F-Minor-Free-Deletion)

| | |
|---|---|
| Constant factor approximation algorithm | [Fomin et al, FOCS 2012] |
| $2^{\mathcal{O}(k)}n^2$ algorithm | [Kim et al, ICALP 2013] |
| $f(\mathcal{F})k^{O(g(\mathcal{F}))}$ kernel | [Fomin et al, FOCS 2012] |
| | |
| | |

# F-Minor-Free-Deletion

- $\mathcal{F}$-Minor-Free Deletion is FPT.
  [Robertson and Seymour, Journal of Combinatorial Theory B]

If $\mathcal{F}$ contains a **planar** graph (Planar F-Minor-Free-Deletion)

| | |
|---|---|
| Constant factor approximation algorithm | [Fomin et al, FOCS 2012] |
| $2^{\mathcal{O}(k)} n^2$ algorithm | [Kim et al, ICALP 2013] |
| $f(\mathcal{F}) k^{O(g(\mathcal{F}))}$ kernel | [Fomin et al, FOCS 2012] |
| $k^{\mathcal{O}(\eta)}$ (Treewidth-$\eta$ Deletion)<br>$k$ is the vertex cover of the input | [Cygan et al, IPEC 2011] |
| | |

# F-Minor-Free-Deletion

- $\mathcal{F}$-Minor-Free Deletion is FPT.
  [Robertson and Seymour, Journal of Combinatorial Theory B]

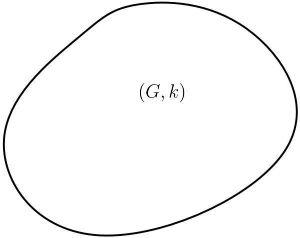
If $\mathcal{F}$ contains a **planar** graph (Planar F-Minor-Free-Deletion)

| | |
|---|---|
| Constant factor approximation algorithm | [Fomin et al, FOCS 2012] |
| $2^{\mathcal{O}(k)}n^2$ algorithm | [Kim et al, ICALP 2013] |
| $f(\mathcal{F})k^{O(g(\mathcal{F}))}$ kernel | [Fomin et al, FOCS 2012] |
| $k^{\mathcal{O}(\eta)}$ (Treewidth-$\eta$ Deletion)<br>$k$ is the vertex cover of the input | [Cygan et al, IPEC 2011] |
| $k^{\mathcal{O}(\Delta(\mathcal{F}))}$ kernel<br>$k$ is the vertex cover of the input | [Fomin et al, JCSS 2014] |

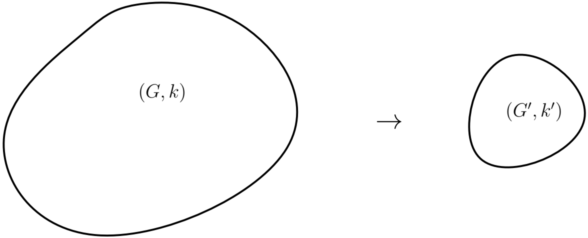
# Open problem

Can we find a kernel for PLANAR F-MINOR-FREE DELETION of size $f(\mathcal{F})k^c$?

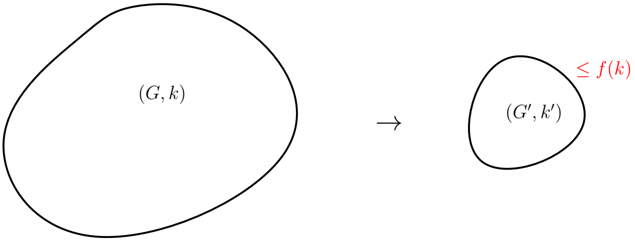
# Kernelization



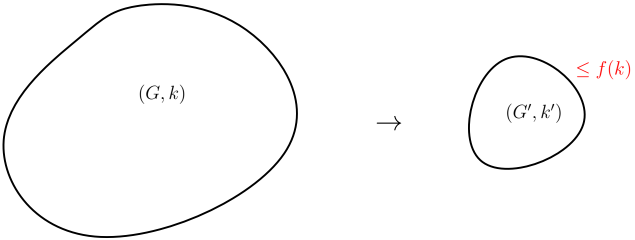
# Kernelization



# Kernelization



# Kernelization




## Kernelization

A *kernelization algorithm* is a polynomial time algorithm that given as input an instance $(G, k)$ to the problem outputs an equivalent instance $(G', k')$ with $k' \leq f(k)$ and $|V(G')| + |E(G')| \leq f(k)$.

# Our Results (Lower Bounds)

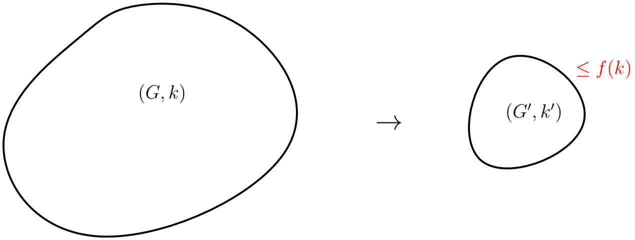
Let $d \geq 3$ be a fixed integer and $\epsilon > 0$.

If the parameterization by solution size $k$ of one of the problems

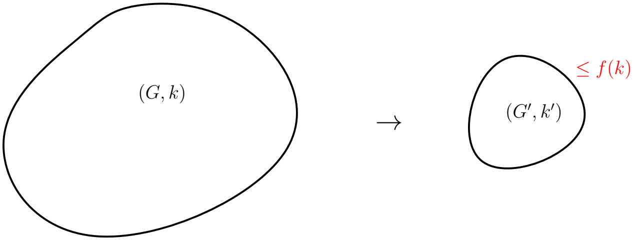
1. $\{K_{d+1}\}$-Minor-Free Deletion,
2. $\{K_{d+1}, P_{4d}\}$-Minor-Free Deletion, and
3. Treewidth-$(d-1)$ Deletion

admits a kernel with $\mathcal{O}(k^{\frac{d}{4}-\epsilon})$ vertices, then NP $\subseteq$ coNP/poly.

# Compression



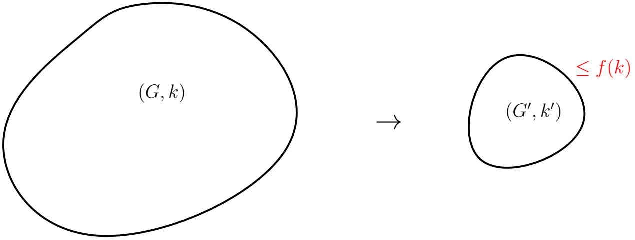
# Compression




## Compression

A *compression algorithm* is a polynomial time algorithm that given as input an instance $(G, k)$ of a problem $P$ outputs an equivalent instance $(G', k')$ of a problem $Q$ with $k' \leq f(k)$ and $|V(G')| + |E(G')| \leq f(k)$.

# Degree-$d$ polynomial-parameter transformation




## Degree-$d$ polynomial-parameter transformation

A *degree-d polynomial-parameter transformation* is a polynomial time algorithm that given as input an instance $(G, k)$ of a problem $P$ outputs an equivalent instance $(G', k')$ of a problem $Q$ with $k' \in \mathcal{O}(k^d)$.

# Exact $d$-Uniform Set Cover

Given:

- A finite set $U$ of size $n$,
- an integer $k$,
- a family $\mathcal{F} \subseteq 2^U$ of sets of size $d$

Asked:

- $\mathcal{F}' \subseteq \mathcal{F}$ of $k$ sets such that each element of $U$ is contained in exactly one set of $\mathcal{F}'$ .

**Theorem (Dell and Marx, SODA 2012)**

*For every fixed $d \geq 3$ and $\epsilon > 0$, there is no compression of size $\mathcal{O}(k^{d-\epsilon})$ for* Exact $d$-Uniform Set Cover, *unless* NP $\subseteq$ coNP/poly.

# EXACT $d$-UNIFORM SET COVER

Given:

- A finite set $U$ of size $n$,
- an integer $k$,
- a family $\mathcal{F} \subseteq 2^U$ of sets of size $d$

Asked:

- $\mathcal{F}' \subseteq \mathcal{F}$ of $k$ sets such that each element of $U$ is contained in exactly one set of $\mathcal{F}'$ (non trivial only when $k = n/d$).

**Theorem (Dell and Marx, SODA 2012)**

*For every fixed $d \geq 3$ and $\epsilon > 0$, there is no compression of size $\mathcal{O}(k^{d-\epsilon})$ for* EXACT $d$-UNIFORM SET COVER, *unless* NP $\subseteq$ coNP/poly.

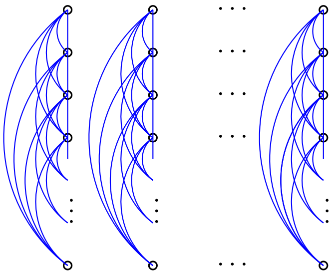
# The reduction

$U = \{1, 2, \ldots, n\}$ and $\mathcal{F}$. We construct $G$ :

$$\begin{array}{cccc}
\circ\, v_{1,1} & \circ & \cdots & \circ\, v_{1,k} \\
\circ\, v_{2,1} & \circ & \cdots & \circ\, v_{2,k} \\
\circ\, v_{3,1} & \circ & \cdots & \circ\, v_{3,k} \\
\circ\, v_{4,1} & \circ & \cdots & \circ\, v_{4,k} \\
\vdots & \vdots & & \vdots \\
\circ\, v_{n,1} & \circ & \cdots & \circ\, v_{n,k}
\end{array}$$

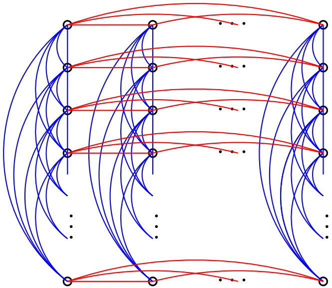
# The reduction

$U = \{1, 2, \ldots, n\}$ and $\mathcal{F}$. We construct $G$ :



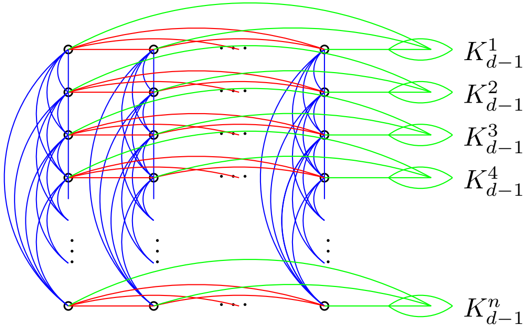
# The reduction

$U = \{1, 2, \ldots, n\}$ and $\mathcal{F}$. We construct $G$ :



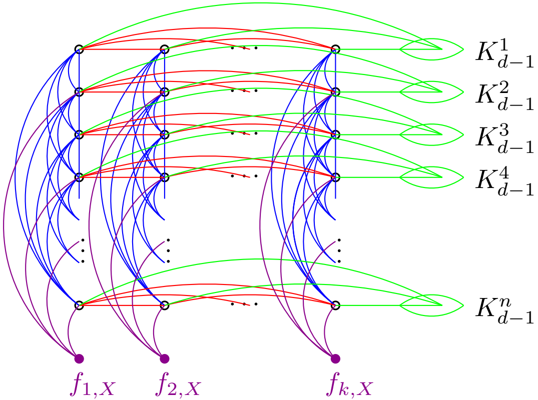
# The reduction

$U = \{1, 2, \ldots, n\}$ and $\mathcal{F}$. We construct $G$ :



# The reduction

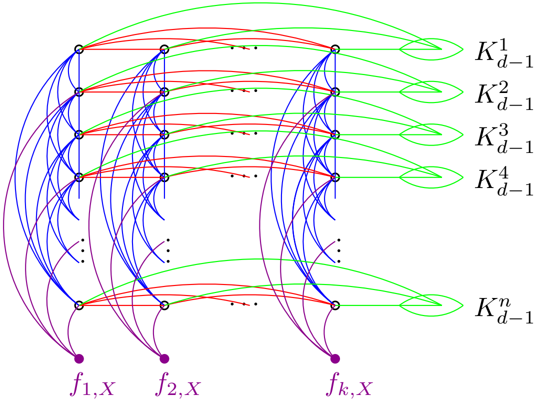
$U = \{1, 2, \ldots, n\}$ and $\mathcal{F}$. We construct $G$ :




for every $X \in \binom{n}{d} \setminus \mathcal{F}$.

# The reduction

$U = \{1, 2, \ldots, n\}$ and $\mathcal{F}$. We construct $G$ :



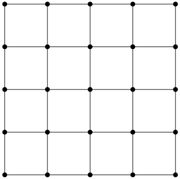

for every $X \in \binom{n}{d} \setminus \mathcal{F}$. Finally, let $k' = k(n-d)$.

# Our Results (TREEDEPTH-$\eta$ DELETION)

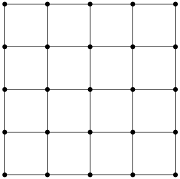
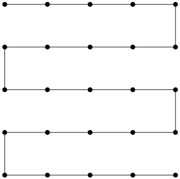
**Theorem**

*For each fixed $\eta$, TREEDEPTH-$\eta$-DELETION has a polynomial kernel with $\mathcal{O}(k^6)$ vertices: an instance $(G, k)$ can be efficiently reduced to an equivalent instance $(G', k)$ with $2^{\mathcal{O}(\eta^2)} k^6$ vertices.*

# Why look into TREEDEPTH-$\eta$-DELETION?

| Excluded Graph | Bounded Parameter |
|---|---|
|  | Treewidth |

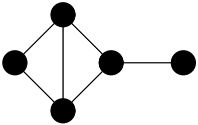
# Why look into TREEDEPTH-$\eta$-DELETION?

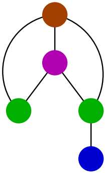
| Excluded Graph | Bounded Parameter |
|---|---|
|  | Treewidth |
|  | Treedepth |

# Treedepth

## Definition

**td**$(G) \leq k$ if and only if there exists a rooted forest $F$ of height at most k such that $G \subseteq$ **clos**$(F)$.





The closure of a rooted forest $F$ is obtained the graph **clos**$(F)$ obtained from $F$ after adding edges between each vertex and its proper ancestors.

# Properties of Treedepth

- $\mathcal{D}_{\eta}$ is minor-closed.

- The class $\mathcal{F}_{\eta} = \mathbf{obs}(\mathcal{D}_{\eta})$ consisting of the minor-obstructions of $\mathcal{D}_{\eta}$ is finite. Thus TREEDEPTH-$\eta$ DELETION is equivalent to $\mathcal{F}_{\eta}$-MINOR-FREE DELETION.

- In particular, $P_{2^{\eta}} \in \mathcal{F}_{\eta}$ and TREEDEPTH-$\eta$ DELETION is equivalent to PLANAR $\mathcal{F}_{\eta}$-MINOR-FREE DELETION.

# Reduction Rule 1 (Vertex Removal)

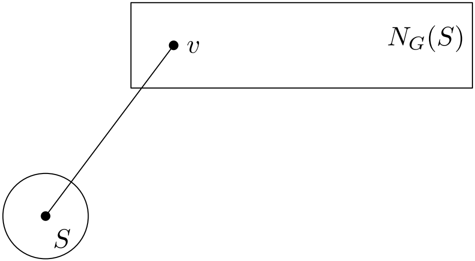
## Lemma

*Let $(G, k)$ be an instance of* Treedepth-$\eta$ Deletion *and let $\ell$ be an integer. Let $S \subseteq V(G)$ such that $N_G(S)$ is a clique and $\mathbf{td}(G[S]) \leq \eta$. For every $v \in N_G(S)$, let $X_1^v, \ldots, X_{\ell+\eta}^v \subseteq V(G)$ induce connected subgraphs of $G$ such that:*

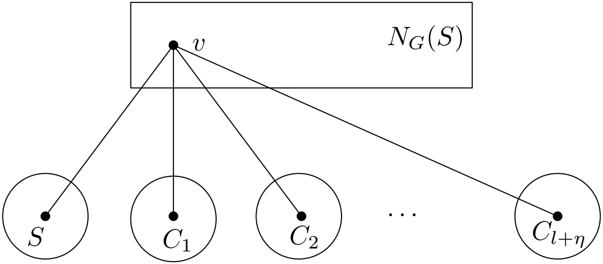
1. $\forall v \in N_G(S), \forall i \in [\ell + \eta]$: $\mathbf{td}(G[X_i^v]) \geq \mathbf{td}(G[S])$ *and* $v \in N_G(X_i^v)$,
2. $\forall v \in N_G(S)$, *the sets* $X_1^v, \ldots, X_{\ell+\eta}^v$ *are pairwise disjoint and disjoint from $S$, and*
3. $G - S$ *has a minimum treedepth-$\eta$ modulator containing $\leq \ell$ vertices of $\mathcal{X}$,*

*where $\mathcal{X} := \bigcup_{v \in N_G(S)} \bigcup_{i \in [\ell+\eta]} X_j^v$. Then $(G, k)$ is equivalent to the instance $(G - S, k)$.*

# Reduction Rule 1 (Vertex Removal)

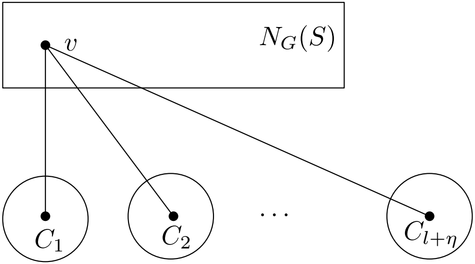


# Reduction Rule 1 (Vertex Removal)




$\mathbf{td}(C_i) \geq \mathbf{td}(S)$

# Reduction Rule 1 (Vertex Removal)




$$\mathbf{td}(C_i) \geq \mathbf{td}(S)$$

# Reduction Rule 2 (Edge Addition)

**Lemma**

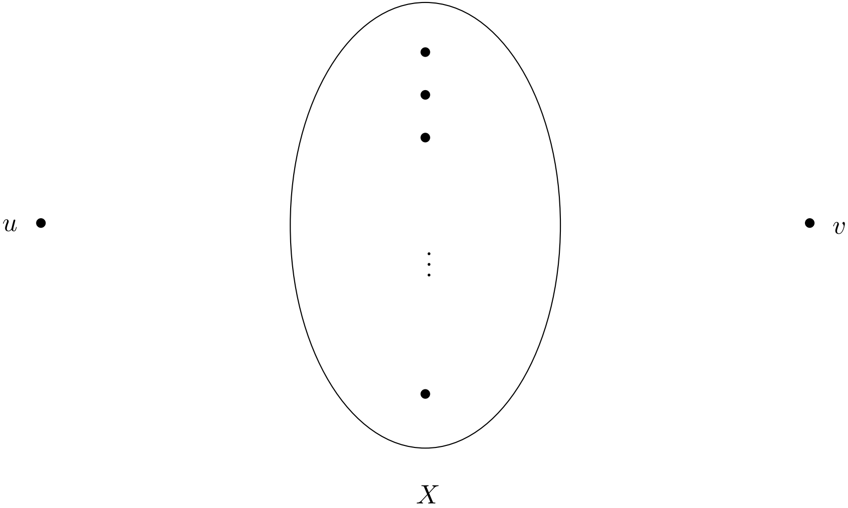
*Let $(G, k)$ be an instance of* TREEDEPTH-$\eta$ DELETION *and let $\ell$ be an integer. Let $X \subseteq V(G)$ and let $\{u, v\} \in \binom{V(G)}{2} \setminus E(G)$. If the following conditions hold:*

1. *the graph $G[X \cup \{u, v\}]$ contains at least $\ell + \eta$ internally vertex-disjoint paths between $u$ and $v$, and*
2. *$G$ has a minimum treedepth-$\eta$ modulator containing $\leq \ell$ vertices of $X$,*

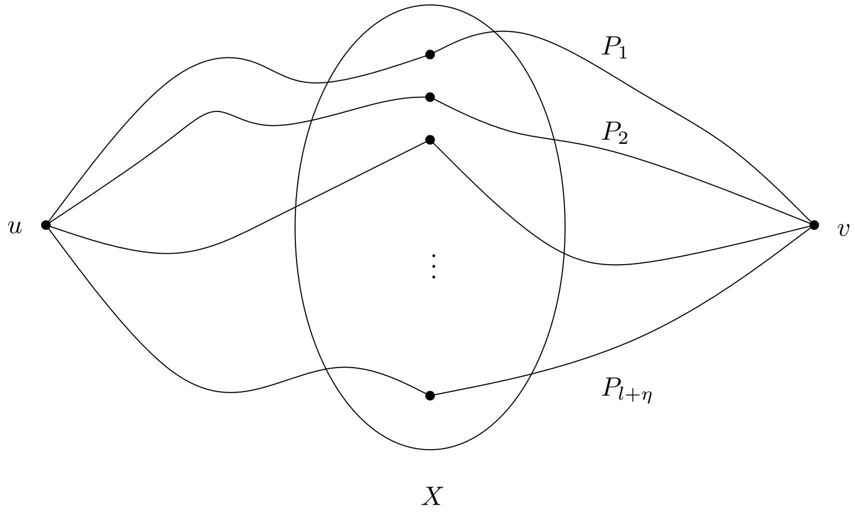
*then $(G, k)$ is equivalent to the instance $(G + uv, k)$ obtained by adding the edge $uv$.*

# Reduction Rule 2 (Edge Addition)

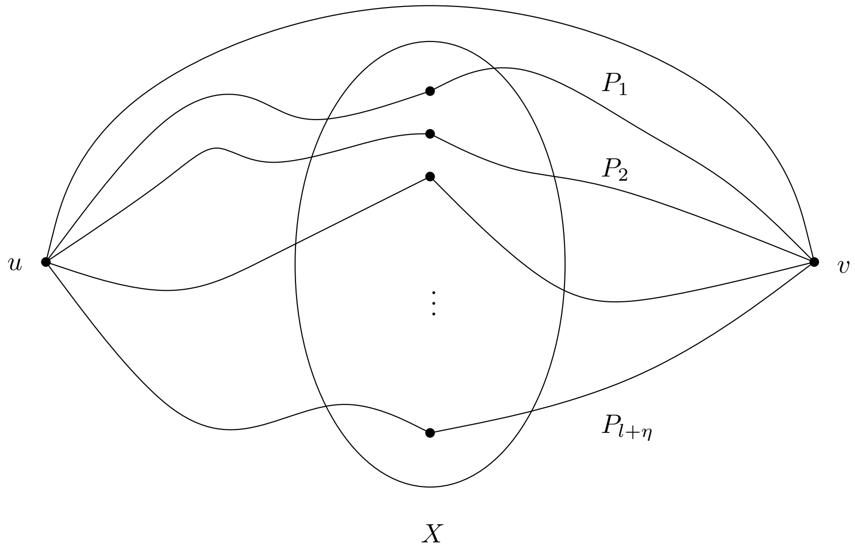
# Reduction Rule 2 (Edge Addition)



# Reduction Rule 2 (Edge Addition)



# Reduction Rule 2 (Edge Addition)



# Reduction Rule 3 (Edge removal)

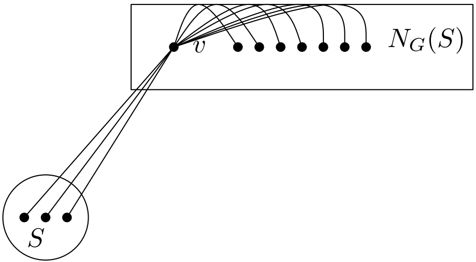
**Lemma**

*Let $(G, k)$ be an instance of* TREEDEPTH-$\eta$ DELETION *and let $\ell$ be an integer. Let $S \subseteq V(G)$ and let $v \in V(G) \setminus S$ such that $N_G(S) \subseteq N_G[v]$. Let $X_1, \ldots, X_{\ell+\eta} \subseteq V(G)$ be connected subgraphs of $G$ such that:*

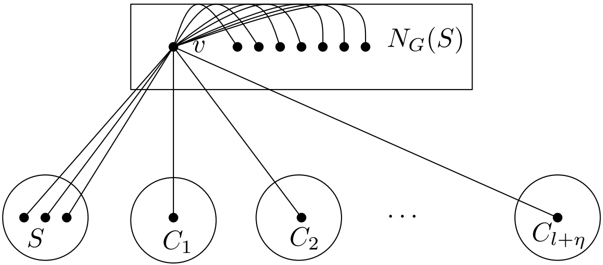
1. $\forall i \in [\ell + \eta]$: $\mathbf{td}(G[X_i]) \geq \mathbf{td}(G[S])$ *and* $v \in N_G(X_i)$,
2. *the sets $X_1, \ldots, X_{\ell+\eta}$ are pairwise disjoint and disjoint from $S$, and*
3. *any graph obtained from $G$ by removing edges between $v$ and $S$ has a minimum treedepth-$\eta$ modulator containing $\leq \ell$ vertices of $\mathcal{X}$,*

*where $\mathcal{X} := \bigcup_{i \in [\ell+\eta]} X_j$. Then $(G, k)$ is equivalent to the instance $(G', k)$, where $G'$ is obtained from $G$ by removing all edges between $v$ and $S$.*

# Reduction Rule 3 (Edge removal)

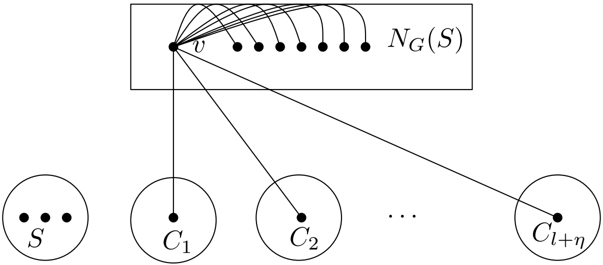


# Reduction Rule 3 (Edge removal)




$\mathbf{td}(C_i) \geq \mathbf{td}(S)$

# Reduction Rule 3 (Edge removal)




$\mathbf{td}(C_i) \geq \mathbf{td}(S)$

# Approximation Algorithm

Together with the Reduction Rules we need the following algorithm to obtain the kernel for $\textsc{Treedepth-}\eta$ $\textsc{Deletion}$.

**Lemma**

*Fix $\eta \in \mathbb{N}$. Given a graph $G$, one can in polynomial time compute a subset $S \subseteq V(G)$ such that $\mathbf{td}(G - S) \leq \eta$ and $|S|$ is at most $2^\eta$ times the size of an optimal treedepth-$\eta$ modulator of $G$.*

[Gajarsky et al., ESA 2013]

# Overview

The kernel is obtained in two phases:

- Decomposition of $(G, k)$ to an equivalent instance $(G', k')$
- Application of Reduction Rules to obtain an equivalent instance $(G'', k)$ of reduced size.

## Graph Decomposition

Let $(G, k)$ be an of Treedepth-$\eta$ Deletion.

# Graph Decomposition

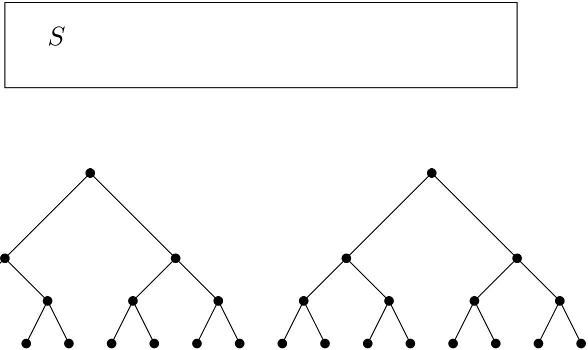
Let $(G, k)$ be an of $\textsc{Treedepth-}\eta$ $\textsc{Deletion}$.

- For every $p, q \in V(G)$ where $\{p, q\} \notin E(G)$, if there exist $k + \eta$ internally disjoint $(p, q)$-paths add $\{p, q\} \in E(G)$.

# Graph Decomposition

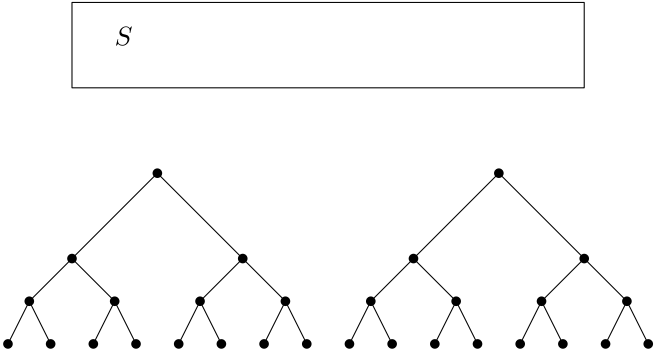
Let $(G, k)$ be an of TREEDEPTH-$\eta$ DELETION.

- For every $p, q \in V(G)$ where $\{p, q\} \notin E(G)$, if there exist $k + \eta$ internally disjoint $(p, q)$-paths add $\{p, q\} \in E(G)$.
- Using the approximation algorithm to find $S$ such that $\mathbf{td}(G \setminus S) \leq \eta$. Then $|S| \leq 2^{\eta}k$. Let $F$ denote the forest where $G \setminus S \subseteq \mathbf{clos}(F)$.



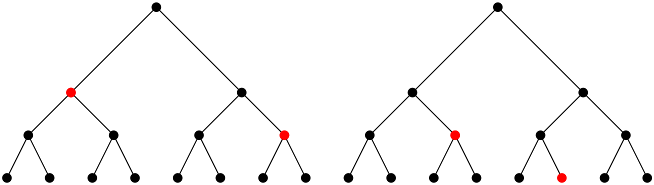
# Graph Decomposition

- For every $p, q \in S$ where $\{p, q\} \notin E(G)$ find a minimum $(p, q)$-separator $Y_{p,q}$ with $|Y_{p,q}| < k + \eta$.



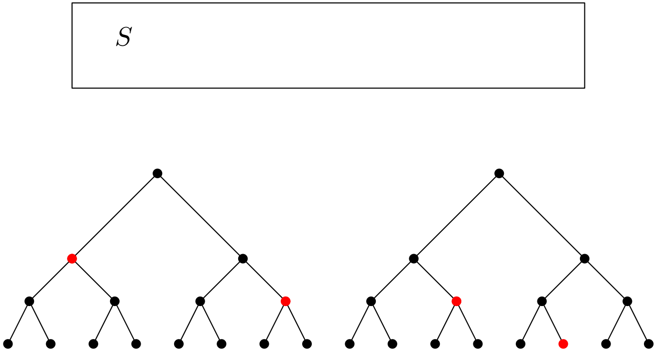
# Graph Decomposition

- For every $p, q \in S$ where $\{p, q\} \notin E(G)$ find a minimum $(p, q)$-separator $Y_{p,q}$ with $|Y_{p,q}| < k + \eta$.

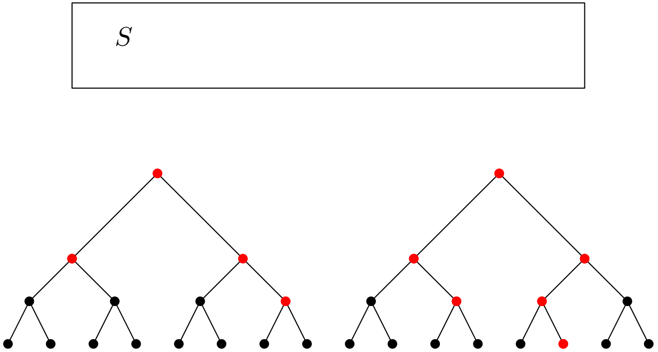
# Graph Decomposition

- Let $Y = \bigcup_{\{p,q\} \notin E(G)} Y_{p,q}$. For every $y \in Y$ add in $Y$ all the proper ancestors of $y$.



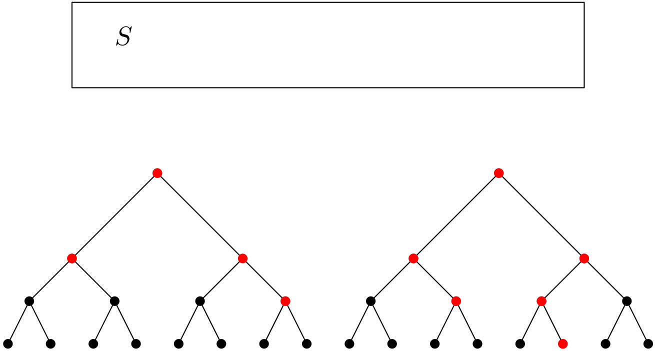
# Graph Decomposition

- Let $Y = \bigcup_{\{p,q\} \notin E(G)} Y_{p,q}$. For every $y \in Y$ add in $Y$ all the proper ancestors of $y$.



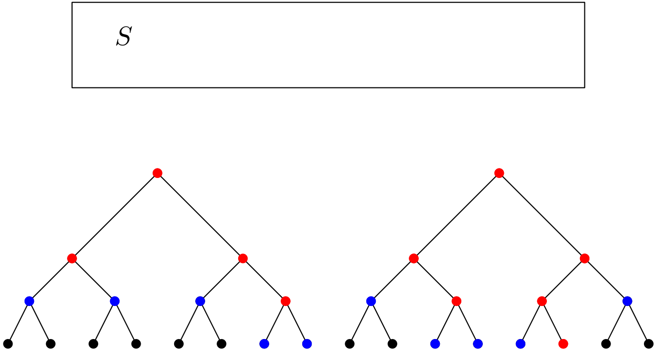
# Graph Decomposition

- Let $\mathcal{T}$ be the roots of the forest $F'$ obtained from $F$ after removing $Y$.



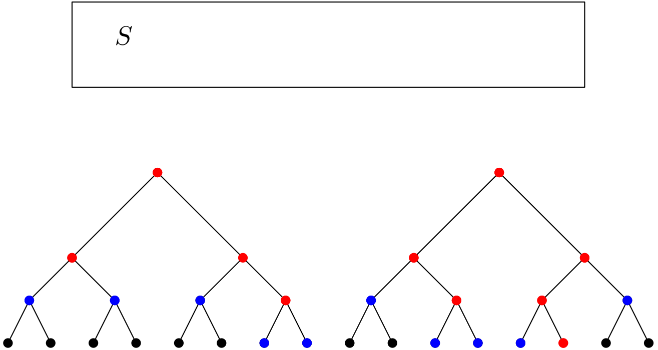
# Graph Decomposition

- Let $\mathcal{T}$ be the roots of the forest $F'$ obtained from $F$ after removing $Y$.



# Graph Decomposition

- While there is a vertex $u_0 \in \mathcal{T}$, where for the set $V(F_{u_0})$ Reduction Rule 1 is applicable, apply Reduction Rule 1 and remove $u_0$ from $\mathcal{T}$.



# Properties of the New Instance

Let $(G, k)$ be an instance of TREEDEPTH-$\eta$ DELETION.

1. $(G, k)$ is equivalent to $(G', k)$
2. $|S| \leq 2^{\eta} \cdot k$.
3. $|Y| \leq \eta(2^{\eta} \cdot k)^2 \cdot (k + \eta)$.
4. For every $u \in V(F') \setminus Y$ the graph $G'[F'_u]$ is connected.

# Properties of the New Instance

Let $(G, k)$ be an instance of TREEDEPTH-$\eta$ DELETION.

1. Let $\mathcal{T}' := \{u \in F' - Y \mid u \text{ is a root or } \pi(u) \in Y\}$. The vertex sets of the connected components of $G' - (S \cup Y)$ are exactly the vertex sets of the subtrees of $F'$ rooted at members of $\mathcal{T}'$.
2. For every connected component $C$ of $G' - (S \cup Y)$, the set $N_{G'}(C) \cap S$ is a clique and for every minimal treedepth-$\eta$ modulator $Z$, $Z \cap V(C) \leq 2\eta$.
3. The number of connected components of $G' - (S \cup Y)$ is at most $(|S| + |Y| + |S|^2 + |S| \cdot |Y| + \eta \cdot |Y|) \cdot (\eta + k)$.

# Bounding the Size of the Connected Components

Let $T$ be the tree in $F'$ with $v$ as root.

- While there are $p, q \in N(T_v) \cup \{v\}$ with $pq \notin E(G)$, joined by $3\eta$ internally vertex disjoint paths in $V(T_v) \cup \{p, q\}$, add $pq \in E(G)$. [Edge addition]

# Bounding the Size of the Connected Components

Let $T$ be the tree in $F'$ with $v$ as root.

- While there are $p, q \in N(T_v) \cup \{v\}$ with $pq \notin E(G)$, joined by $3\eta$ internally vertex disjoint paths in $V(T_v) \cup \{p, q\}$, add $pq \in E(G)$. [Edge addition]

- While there exist distinct children $c_0, \ldots, c_{3\eta}$ of $v$ s.t. $s \in N_S(c_0) \neq \emptyset$ and $N_G[T_{c_0}] \subseteq N_G[s]$, $\mathbf{td}(G[T_{c_i}]) \geq G[T_{c_0}]$ and $s \in N_G[T_{c_i}]$, remove the edges $vu$, $u \in V(T_{c_0})$ [Edge Deletions]

# Bounding the Size of the Connected Components

Let $T$ be the tree in $F'$ with $v$ as root.

- While there are $p, q \in N(T_v) \cup \{v\}$ with $pq \notin E(G)$, joined by $3\eta$ internally vertex disjoint paths in $V(T_v) \cup \{p, q\}$, add $pq \in E(G)$. [Edge addition]

- While there exist distinct children $c_0, \ldots, c_{3\eta}$ of $v$ s.t. $s \in N_S(c_0) \neq \emptyset$ and $N_G[T_{c_0}] \subseteq N_G[s]$, $\mathbf{td}(G[T_{c_i}]) \geq G[T_{c_0}]$ and $s \in N_G[T_{c_i}]$, remove the edges $vu$, $u \in V(T_{c_0})$ [Edge Deletions]

- While there exists a child $c^*$ of $v$ s.t. $N_G[T_{c^*}]$ is a clique, and for every $w \in N_G[T_{c^*}]$ there are $3\eta$ distinct children of $v$, $c_i^w \neq c^*$, $i \in [3\eta]$ with $\mathbf{td}(G[T_{c_i^w}]) \geq \mathbf{td}(G[T_{c^*}])$ and $w \in N_G[T_{c_i^w}]$, remove $T_{c^*}$ from $F$ and from $G$. [Vertex deletions]

# Bounding the Size of the Connected Components

Let $T$ be the tree in $F'$ with $v$ as root.

- While there are $p, q \in N(T_v) \cup \{v\}$ with $pq \notin E(G)$, joined by $3\eta$ internally vertex disjoint paths in $V(T_v) \cup \{p, q\}$, add $pq \in E(G)$. [Edge addition]

- While there exist distinct children $c_0, \ldots, c_{3\eta}$ of $v$ s.t. $s \in N_S(c_0) \neq \emptyset$ and $N_G[T_{c_0}] \subseteq N_G[s]$, $\mathbf{td}(G[T_{c_i}]) \geq G[T_{c_0}]$ and $s \in N_G[T_{c_i}]$, remove the edges $vu$, $u \in V(T_{c_0})$ [Edge Deletions]

- While there exists a child $c^*$ of $v$ s.t. $N_G[T_{c^*}]$ is a clique, and for every $w \in N_G[T_{c^*}]$ there are $3\eta$ distinct children of $v$, $c_i^w \neq c^*$, $i \in [3\eta]$ with $\mathbf{td}(G[T_{c_i^w}]) \geq \mathbf{td}(G[T_{c^*}])$ and $w \in N_G[T_{c_i^w}]$, remove $T_{c^*}$ from $F$ and from $G$. [Vertex deletions]

- Call the algorithm for every remaining child of $v$.

# The Size of the Reduced Components

By counting arguments that depend on the properties of the connected components

$$|C| \leq \eta \cdot (2 \cdot 3\eta \cdot 2^{\eta})^{\eta}(|S| + 1)$$

This implies that

$$V(G') \in \mathcal{O}(k^6)$$

# Summarizing

- When $\mathcal{F}$ is one the following three graph classes
  - $\{K_{d+1}\}$
  - $\{K_{d+1}, P_{4d}\}$
  - Obstructions to Treewidth-$(d-1)$

  the size of the kernel parameterized by the solution size is $\Omega(k^{\frac{d}{4}} - \epsilon)$ unless NP $\subseteq$ coNP/poly.

- The problem $\textsc{Treedepth-}\eta$ $\textsc{Deletion}$ parameterized by the solution size admits a kernel with $2^{\mathcal{O}(\eta^2)}k^6$ vertices.

# Open Problems

- Does there exist a family $\mathcal{F}$ that does not contain planar graphs and admits a polynomial size kernel?

- Is it possible to obtain a dichotomy theorem characterizing for which families $\mathcal{F}$ the problem Planar $\mathcal{F}$-Minor-Free Deletion admits uniformly polynomial kernels?

- Obtain a kernel for Vertex Planarization.

# Thank you for your attention!