Linear kernels for graphs excluding a topological minor

Alexander Langer, **Felix Reidl**, Somnath Sikdar, Peter Rossmanith

Lehr- und Forschungsgebiet Theoretische Informatik

RWTH AACHEN

January 31, 2012

Contents

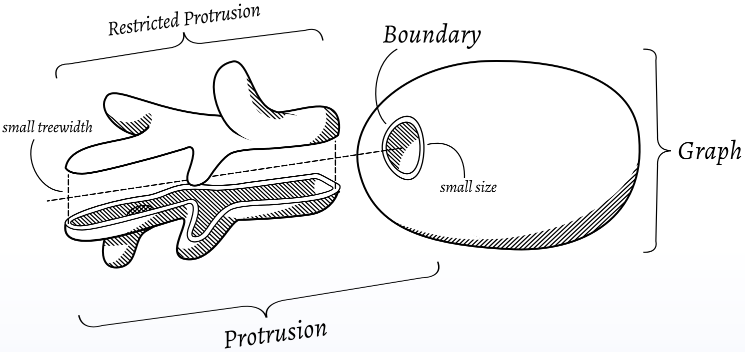
Linear kernels in sparse graphs

Overview

- Framework for planar graphs
  Guo and Niedermeier: *Linear problem kernels for NP-hard problems on planar graphs*
- Meta-result for graphs of bounded genus
  Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh and Thilikos: *(Meta) Kernelization*
- Meta-result for graphs excluding a fixed graph as a minor
  Fomin, Lokshtanov, Saurabh and Thilikos: *Bidimensionality and kernels*
- *Our contribution*: general result for graphs excluding a fixed graph as a *topological* minor

Reduction via protrusions

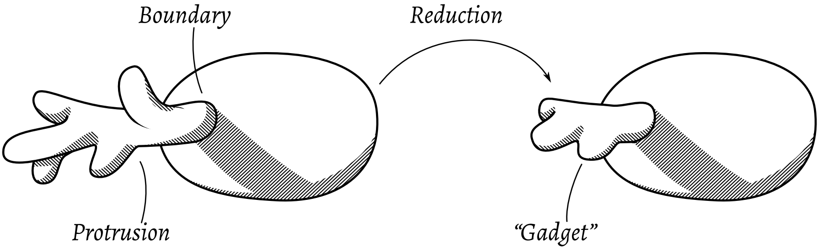
Protrusion anatomy



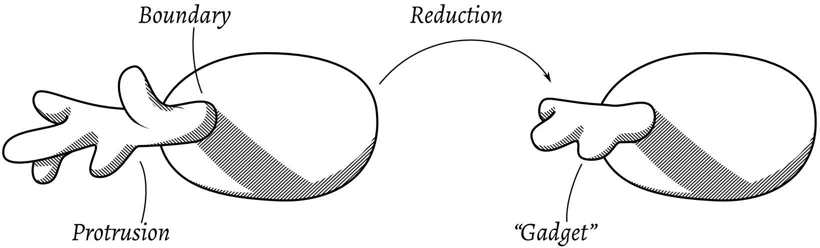
Definition

$X \subseteq V(G)$ is a t-*protrusion* if

1. $|\partial(X)| = |N(X) \setminus X| \leqslant t$ (small boundary)
2. $\mathbf{tw}(G[X]) \leqslant t$ (small treewidth)

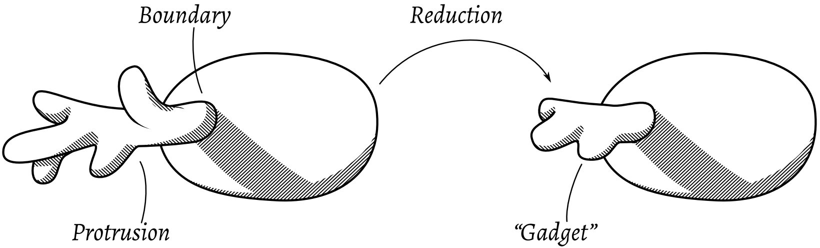


We want to replace a large protrusion by something smaller.
Requires that the problem...



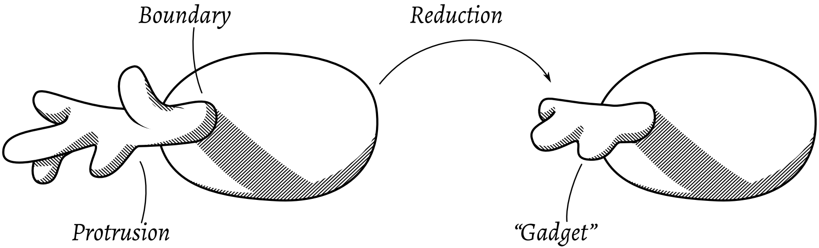
We want to replace a large protrusion by something smaller.
Requires that the problem...

1. ...can be solved by dynamic programming on graphs of bounded treewidth



We want to replace a large protrusion by something smaller.
Requires that the problem...

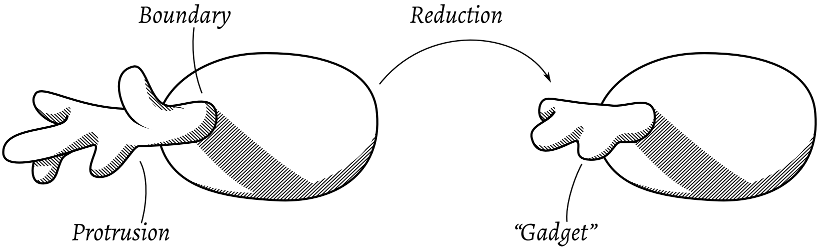
1. ...can be solved by dynamic programming on graphs of bounded treewidth
2. ...admits small gadgets (finite integer index)



We want to replace a large protrusion by something smaller.
Requires that the problem...

1. ...can be solved by dynamic programming on graphs of bounded treewidth
2. ...admits small gadgets (finite integer index)

Note: the reduction can decrease the parameter.



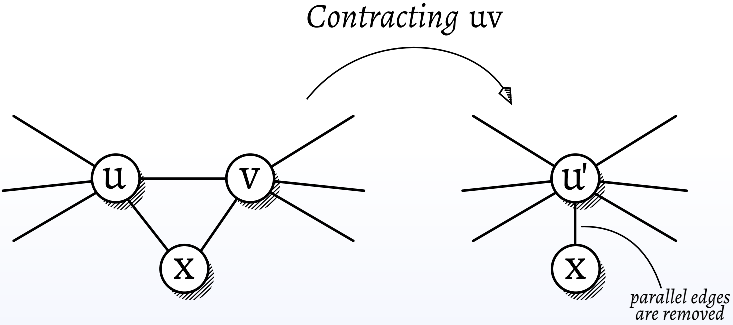
We want to replace a large protrusion by something smaller.
Requires that the problem...

1. ...can be solved by dynamic programming on graphs of bounded treewidth
2. ...admits small gadgets (finite integer index)

Note: the reduction can decrease the parameter.
This is the only reduction.

(Topological) Minors

Edge contraction



Graph relations

(now with contractions!)

Graph relations

(now with contractions!)

Relation	Operations
induced subgraph	delete vertices
subgraph	
topological minor	
minor	

Graph relations

(now with contractions!)

Relation	Operations
induced subgraph	delete vertices
subgraph	delete vertices and edges
topological minor	
minor	

Graph relations

(now with contractions!)

Relation	Operations
induced subgraph	delete vertices
subgraph	delete vertices and edges
topological minor	
minor	delete vertices and edges, contract edges

Graph relations

(now with contractions!)

Relation	Operations
induced subgraph	delete vertices
subgraph	delete vertices and edges
topological minor	delete vertices and edges, contract edges *incident to a degree-2 vertex*
minor	delete vertices and edges, contract edges

Properties of $\mathcal{H}$-topological-minor-free graphs

Let G be a graph excluding H as a topological minor.

Let G be a graph excluding H as a topological minor.

- Not interested in structure of H, but its size $r = |H|$

Let G be a graph excluding H as a topological minor.

- Not interested in structure of H, but its size $r = |H|$
- In particular: K_r not a topological minor of G

Let G be a graph excluding H as a topological minor.

- Not interested in structure of H, but its size $r = |H|$
- In particular: K_r not a topological minor of G

Important properties:

Let G be a graph excluding H as a topological minor.

- Not interested in structure of H, but its size $r = |H|$
- In particular: K_r not a topological minor of G

Important properties:

1. $m \leqslant \frac{1}{2}\beta r^2 n$ (for some $\beta < 10$)

Let G be a graph excluding H as a topological minor.

- Not interested in structure of H, but its size $r = |H|$
- In particular: K_r not a topological minor of G

Important properties:

1. $m \leqslant \frac{1}{2}\beta r^2 n$ (for some $\beta < 10$)
2. #cliques $\leqslant 2^{\tau r \log r} n$ (for some $\tau < 4.51$)

Let G be a graph excluding H as a topological minor.

- Not interested in structure of H, but its size $r = |H|$
- In particular: K_r not a topological minor of G

Important properties:

1. $m \leqslant \frac{1}{2}\beta r^2 n$ (for some $\beta < 10$)
2. #cliques $\leqslant 2^{\tau r \log r} n$ (for some $\tau < 4.51$)
3. Closed under taking topological minors

Our result and how it works

Requirements

(besides the ones mentioned before)

Definition (Treewidth bounding)

A parameterized graph problem Π is called *treewidth bounding* if for every $(G, k) \in \Pi$ it holds that there exists a set $S \subseteq V(G)$ such that

1. $|S| \leqslant ck$
2. $\mathbf{tw}(G - S) \leqslant t$

for constants c, t only depending on Π.

Requirements

(besides the ones mentioned before)

Definition (Treewidth bounding)

A parameterized graph problem Π is called *treewidth bounding* if for every $(G, k) \in \Pi$ it holds that there exists a set $S \subseteq V(G)$ such that

1. $|S| \leqslant ck$
2. $\mathbf{tw}(G - S) \leqslant t$

for constants c, t only depending on Π.

- S usually is the solution set

Requirements
(besides the ones mentioned before)

Definition (Treewidth bounding)

A parameterized graph problem Π is called *treewidth bounding* if for every $(G, k) \in \Pi$ it holds that there exists a set $S \subseteq V(G)$ such that

1. $|S| \leqslant ck$
2. $\mathbf{tw}(G - S) \leqslant t$

for constants c, t only depending on Π.

- S usually is the solution set
- Vertex Cover, Feedback Vertex Set in general graphs

Requirements

(besides the ones mentioned before)

Definition (Treewidth bounding)

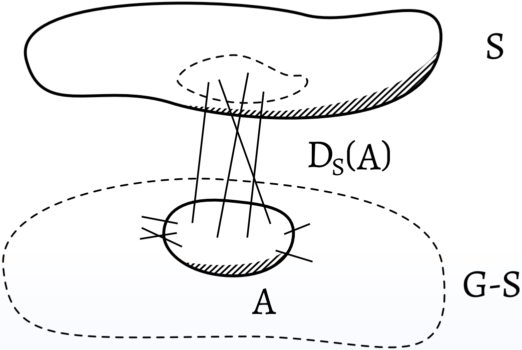
A parameterized graph problem Π is called *treewidth bounding* if for every $(G, k) \in \Pi$ it holds that there exists a set $S \subseteq V(G)$ such that

1. $|S| \leqslant ck$
2. $\mathbf{tw}(G - S) \leqslant t$

for constants c, t only depending on Π.

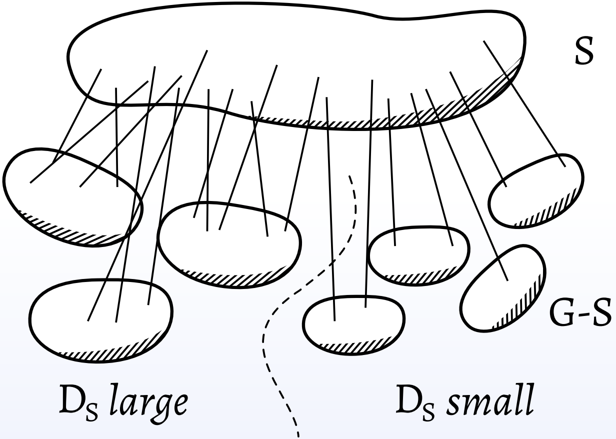
- S usually is the solution set
- Vertex Cover, Feedback Vertex Set in general graphs
- Chordal Vertex Deletion in graphs with bounded clique-size

A little bit of notation

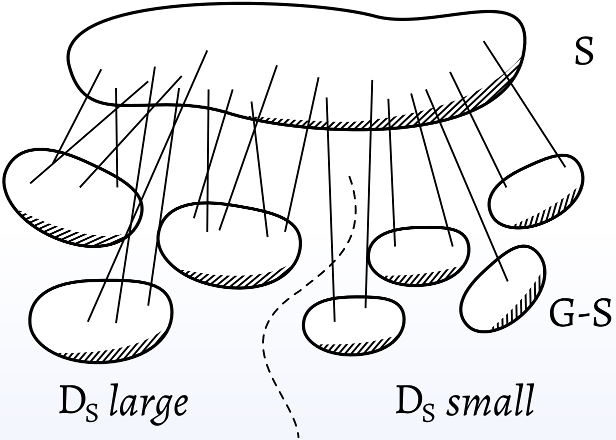


We write $D_S(A) = |\{u \in S \mid v \in A : uv \in E(G)\}|$ for the number of vertices in S that have neighbours in A (for disjoint sets S, A)

A decomposition

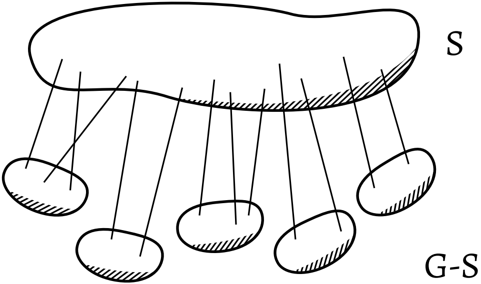


A decomposition



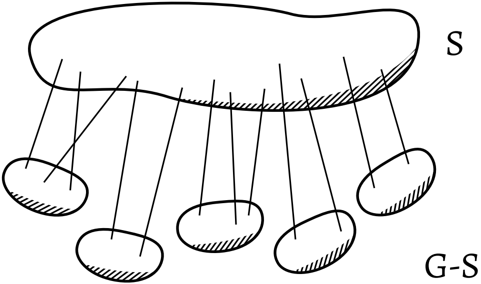
Reduced instance: large protrusions are gone

Small-degree components



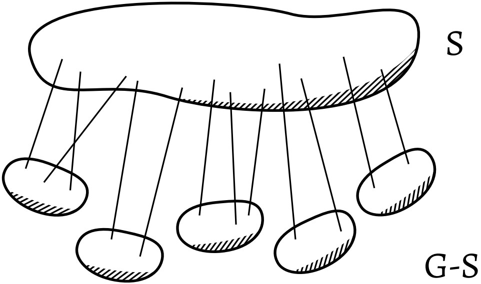
- $D_S(C) < r$, therefore boundary of size r

Small-degree components



- $D_S(C) < r$, therefore boundary of size r
- C has constant treewidth (problem is treewidth bounding)

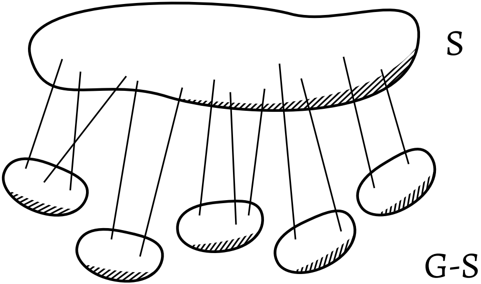
Small-degree components



- $D_S(C) < r$, therefore boundary of size r
- C has constant treewidth (problem is treewidth bounding)

⇒ Each small-degree component has constant size (reduced instance)

Small-degree components

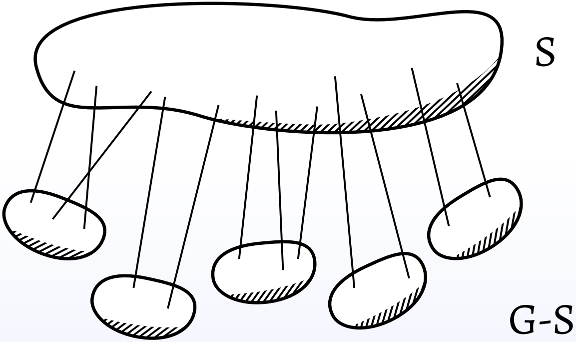


- $D_S(C) < r$, therefore boundary of size r
- C has constant treewidth (problem is treewidth bounding)

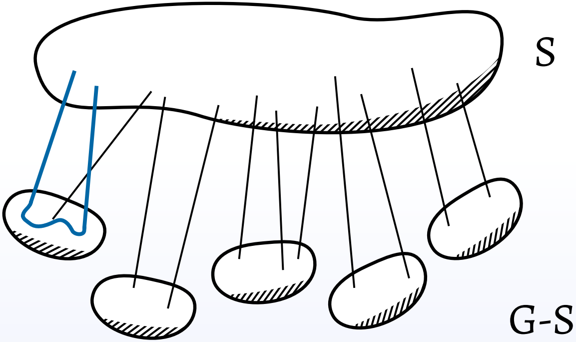
$\Rightarrow$ Each small-degree component has constant size (reduced instance)

- What about the *number* of small-degree components?

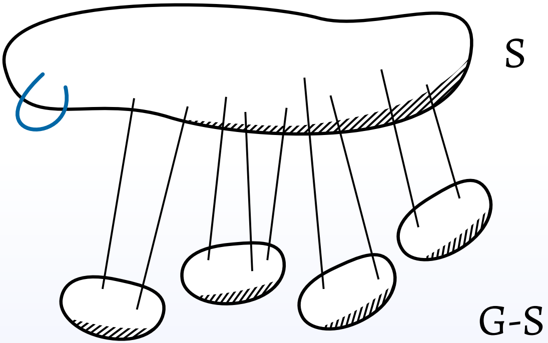
Small-degree components



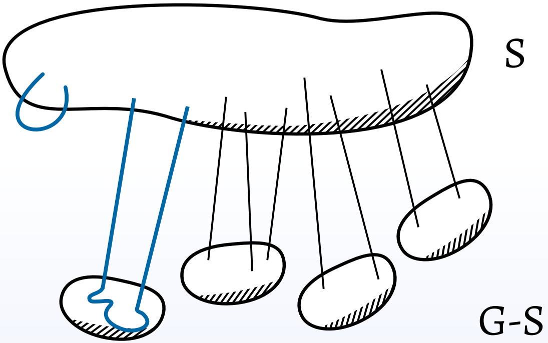
Small-degree components



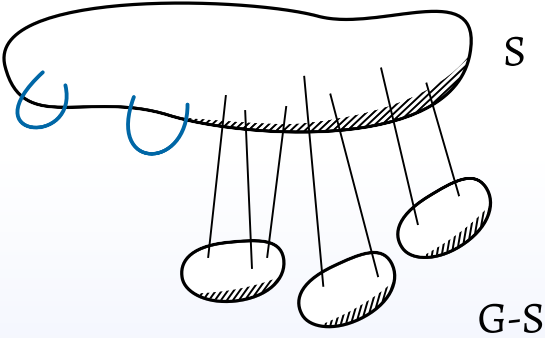
Small-degree components



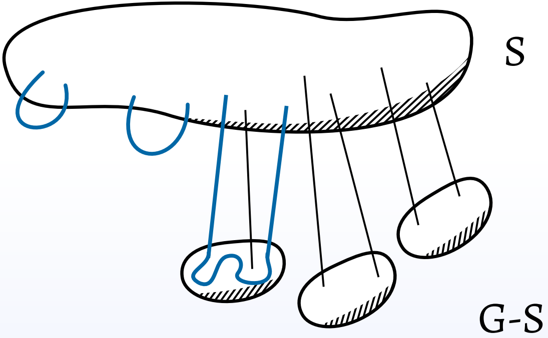
Small-degree components



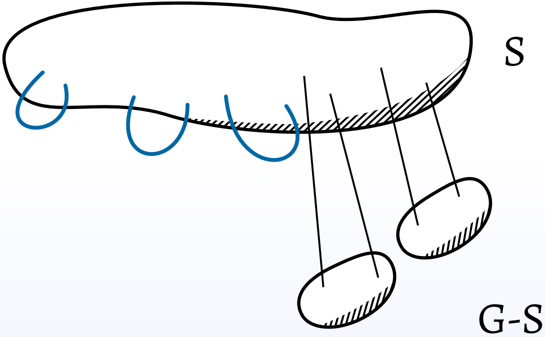
Small-degree components



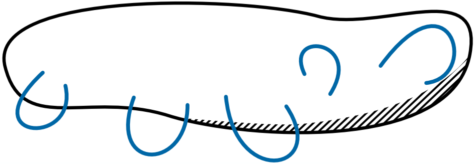
Small-degree components



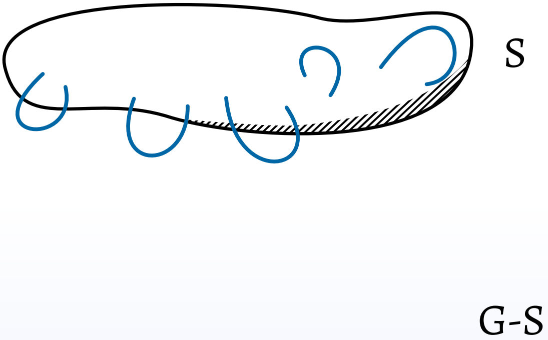
Small-degree components



Small-degree components

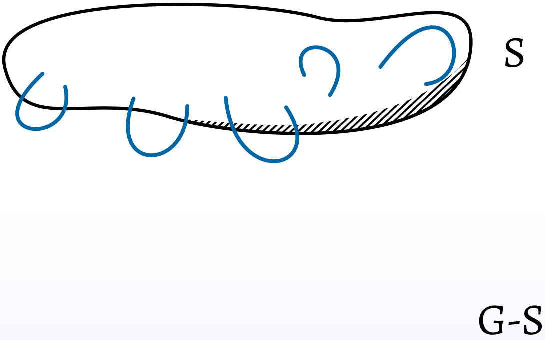


Small-degree components



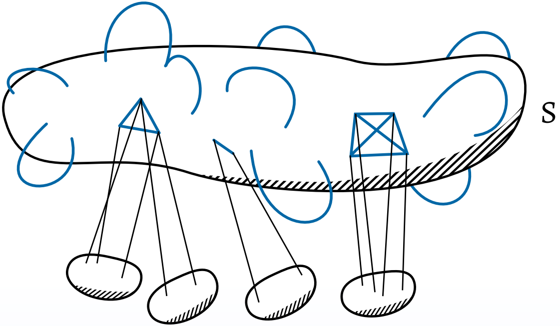
- How often can we do this?

Small-degree components



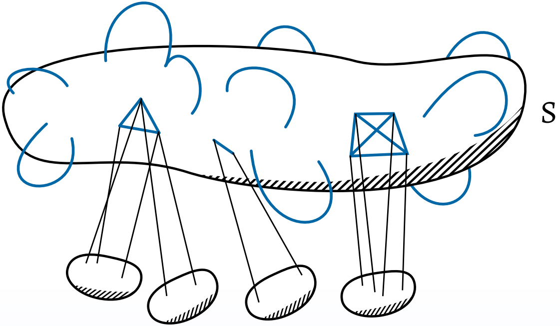
- How often can we do this?
- Is it exhaustive?

Small-degree components



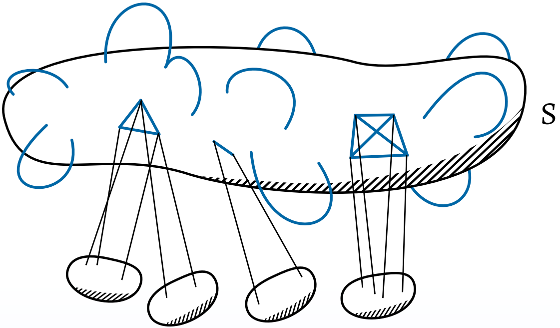
- Components now connected to cliques (or not finished)

Small-degree components



- Components now connected to cliques (or not finished)
- $G[S]$ is H-topological minor free, therefore...

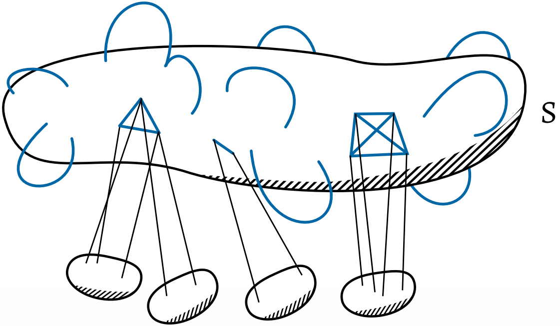
Small-degree components



- Components now connected to cliques (or not finished)
- $G[S]$ is H-topological minor free, therefore...

  ... $O(|S|) = O(k)$ cliques

Small-degree components

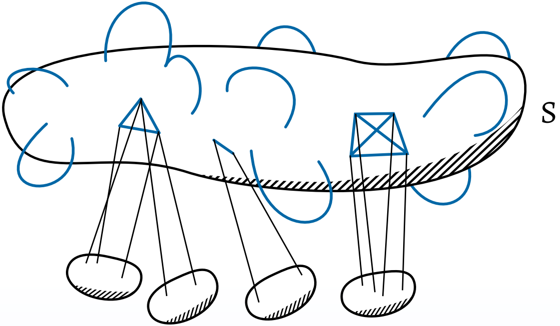


- Components now connected to cliques (or not finished)
- $G[S]$ is H-topological minor free, therefore...

  ... $O(|S|) = O(k)$ cliques

  ... $O(|S|) = O(k)$ edges

Small-degree components



- Components now connected to cliques (or not finished)
- $G[S]$ is H-topological minor free, therefore...
  ... $O(|S|) = O(k)$ cliques
  ... $O(|S|) = O(k)$ edges
- Constant number of vertices in components connected to a common clique (or large protrusion in G)

50% done!

(not)

$O(k)$ vertices in small-degree components

Large-degree components

Very technical. Two ingredients:

Large-degree components

Very technical. Two ingredients:

1. At most $O(k)$ connected *subgraphs* with $D_S \geqslant r$

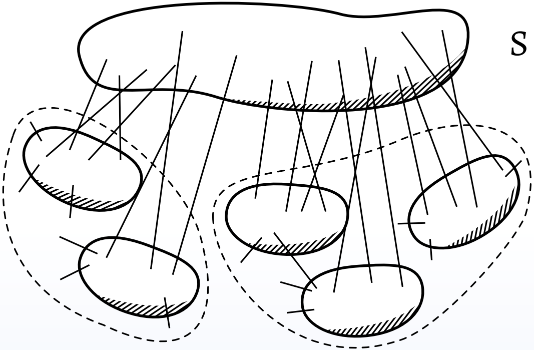
Large-degree components

Very technical. Two ingredients:

1. At most $O(k)$ connected *subgraphs* with $D_S \geqslant r$
2. Tree-decomposition allows us to find many such subgraphs *of constant size*

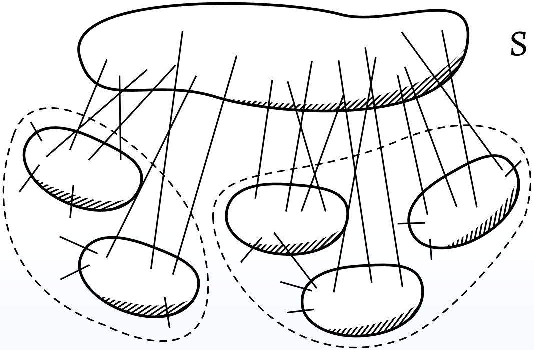
Large-degree components

Ingredient one



Large-degree components

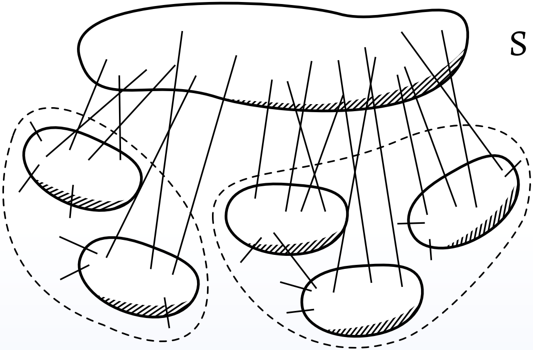
Ingredient one



- Same idea as before: contract connected subgraphs into edges in S

Large-degree components

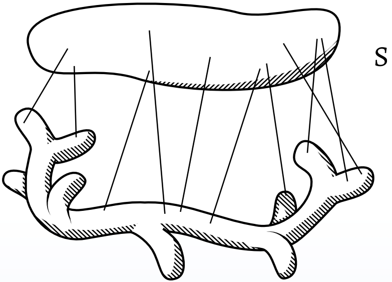
Ingredient one



- Same idea as before: contract connected subgraphs into edges in S
- Exhaustive, else K_r as a subgraph in S and thus H as a topological minor in G

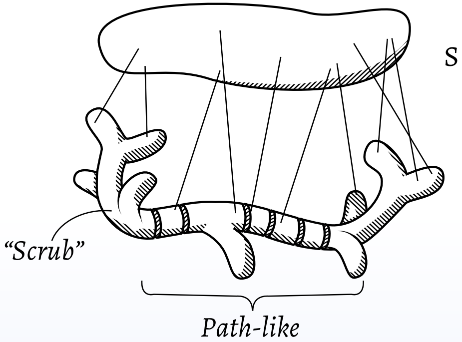
Large-degree components

Ingredient two



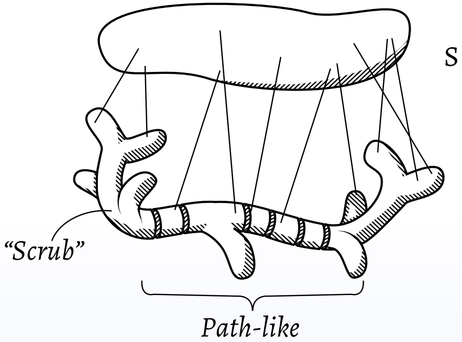
Large-degree components

Ingredient two



Large-degree components

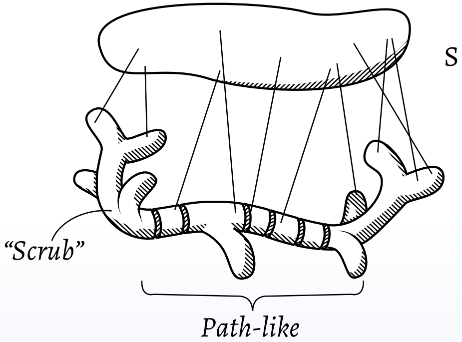
Ingredient two



- Walk along path-decomposition

Large-degree components

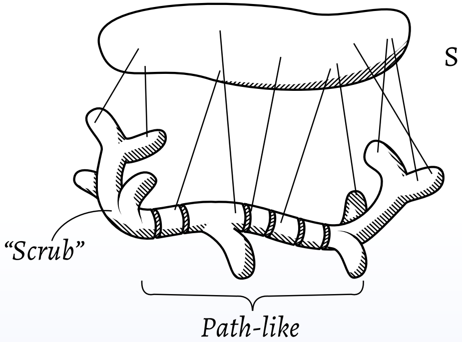
Ingredient two



- Walk along path-decomposition
- Small degree $\Rightarrow$ Small boundary

Large-degree components

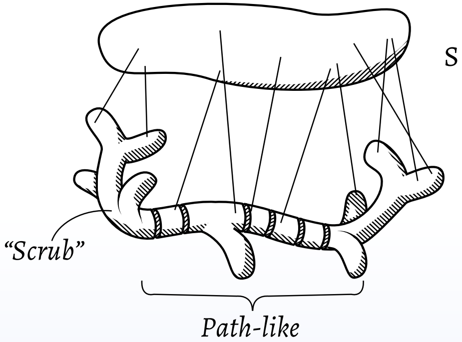
Ingredient two



- Walk along path-decomposition
- Small degree ⇒ Small boundary
- If more than $\varpi(2t + r)$ vertices seen: subgraph has large degree wrt S

Large-degree components

Ingredient two



- Walk along path-decomposition
- Small degree $\Rightarrow$ Small boundary
- If more than $\varpi(2t+r)$ vertices seen: subgraph has large degree wrt S

Conclusion

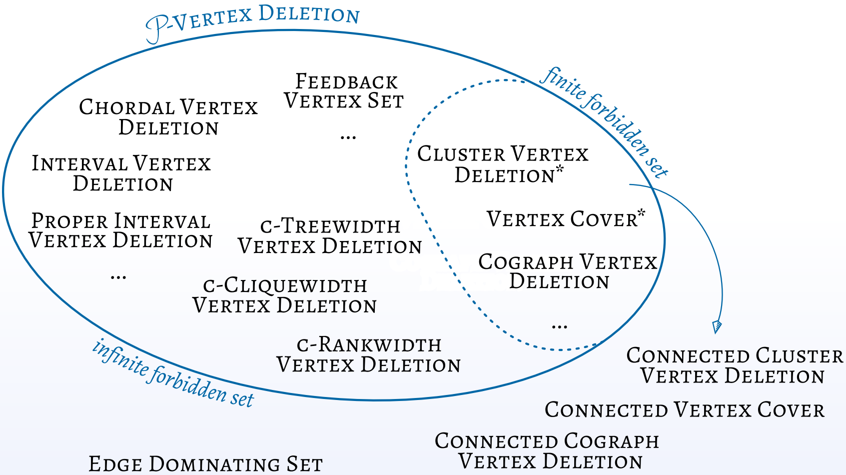
The result

We have shown that problems. . .

1. . . . that can be solved in polynomial time via dynamic programming on graphs of bounded treewidth
2. . . . that have finite integer index
3. . . . and that are treewidth bounding

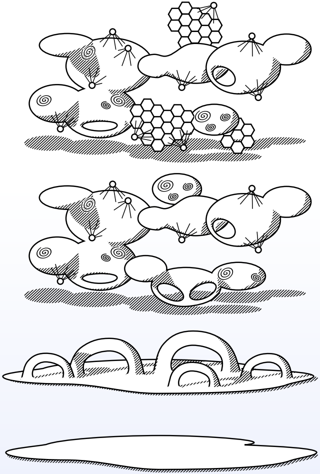
admit linear kernels on graphs excluding a fixed topological minor.

Examples



𝒫-Vertex Deletion
finite forbidden set
infinite forbidden set
Feedback Vertex Set
Chordal Vertex Deletion
...
Interval Vertex Deletion
Proper Interval Vertex Deletion
...
c-Treewidth Vertex Deletion
c-Cliquewidth Vertex Deletion
c-Rankwidth Vertex Deletion
Cluster Vertex Deletion*
Vertex Cover*
Cograph Vertex Deletion
...
Connected Cluster Vertex Deletion
Connected Vertex Cover
Connected Cograph Vertex Deletion
...
Edge Dominating Set

Trade-off: class of instances vs. problem requirements



H-Topological-Minor-Free	*Treewidth-bounding*
∪	
H-Minor-Free	*Bidimensional* *+ separation property*
∪	
Bounded Genus	*Quasi-compact*
∪	
Planar	*“Distance-property”*

Open questions

- What about graphs excluding a fixed *induced minor*/*contraction*/*immersion*? Which other notions of sparse graphs allow such a theorem?
- Can we do this for DOMINATING SET and similar problems? (Grohe & Marx -decomposition!)
- Are there interesting *polynomially* treewidth bounding problems? (We looked at *linear* treewidth bounding)

Open questions

- What about graphs excluding a fixed *induced minor*/*contraction*/*immersion*? Which other notions of sparse graphs allow such a theorem?
- Can we do this for DOMINATING SET and similar problems? (Grohe & Marx -decomposition!)
- Are there interesting *polynomially* treewidth bounding problems? (We looked at *linear* treewidth bounding)

Thank you!