Algorithmic Meta-Theorems

192.122
WS21/22

Jan Dreier
dreier@ac.tuwien.ac.at

ALGORITHMS AND COMPLEXITY GROUP
TU WIEN Informatics

Recap

We learned about Gaifman's locality theorem and used it do find a first-order meta-theorem for locally bounded treewidth.

This captures three natural classes of graphs.

- bounded treewidth
- planar graphs
- bounded degree

However, locally bounded treewidth is not very robust. It is not closed under adding apex-vertices.

Roadmap

We want to find the most general structural notion of “sparsity” that still still admits fpt first-order model-checking.

Roadmap

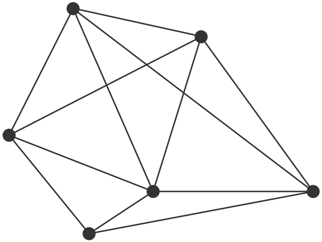
We want to find the most general structural notion of "sparsity" that still still admits fpt first-order model-checking.

If you want to know more after this course, check out lecture notes and recordings of the fantastic sparsity lecture by Marcin Pilipczuk and Michał Pilipczuk.

- lecture notes:
  `mimuw.edu.pl/~mp248287/sparsity2/`
- video recordings:
  `youtube.com/playlist?list=`
  `PLzdZSKerwrXrUPVDx6pHUPNdurKxqC4VD`

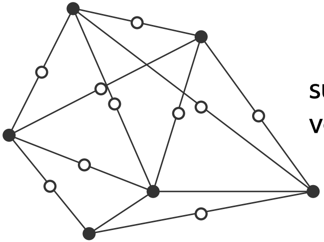
Sparsity Choices

- Every graph is “sparse” if you subdivide the edges.



Sparsity Choices

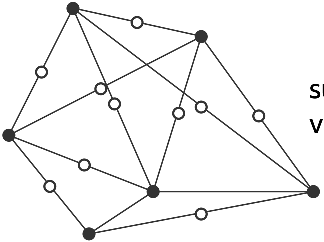
- Every graph is “sparse” if you subdivide the edges.



subdivision adds $1/2$ vertex per edge

Sparsity Choices

- Every graph is “sparse” if you subdivide the edges.

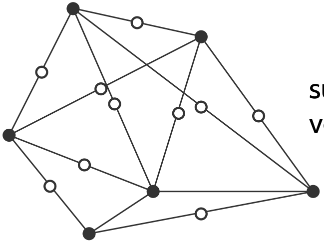


subdivision adds $1/2$ vertex per edge

- Do we consider such subdivisions sparse?
  - Yes: Degeneracy
  - No: Bounded expansion and nowhere dense graph classes

Sparsity Choices

- Every graph is “sparse” if you subdivide the edges.

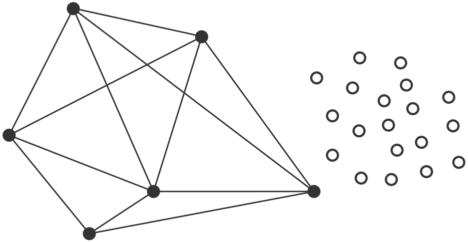


subdivision adds $1/2$ vertex per edge

- Do we consider such subdivisions sparse?
  - Yes: Degeneracy
  - No: Bounded expansion and nowhere dense graph classes
- Let us first consider the case where we say “Yes”.

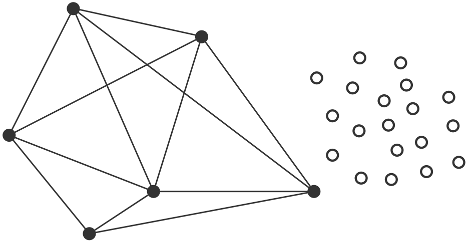
Sparsity Choices

- Are we allowed to make a graph “sparse” by adding extra vertices?



Sparsity Choices

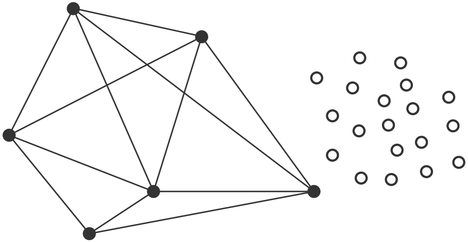
- Are we allowed to make a graph "sparse" by adding extra vertices?



- Definitely no!

Sparsity Choices

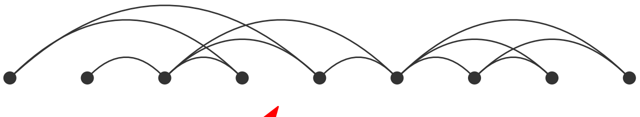
- Are we allowed to make a graph “sparse” by adding extra vertices?



- Definitely no!
- We want a width measure that does not increase if we take a subgraph. It should be “closed under subgraphs“ (hereditary).

Degeneracy

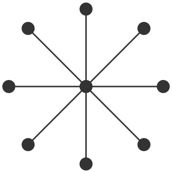
- The *degeneracy* of a a graph is the minimum number k such that there is an ordering of the vertices where no vertex has more than k neighbors on the left.



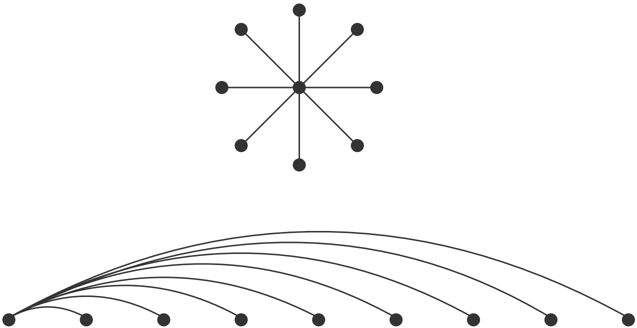
at most two left neighbors

Degeneracy

What is the degeneracy here?

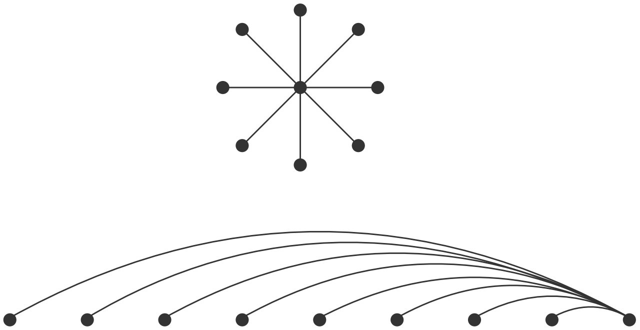


What is the degeneracy here?



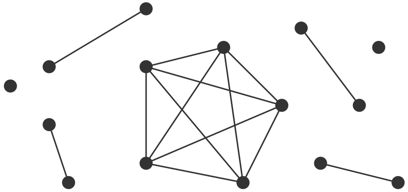
Degeneracy

What is the degeneracy here?



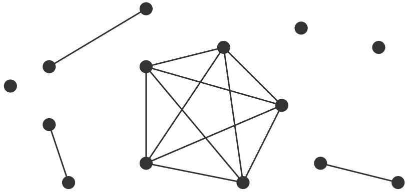
Degeneracy

- Degeneracy is closed under subgraphs.



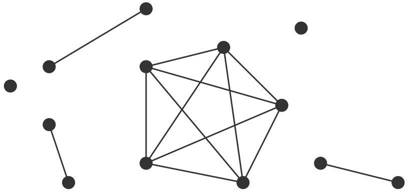
Degeneracy

- Degeneracy is closed under subgraphs.



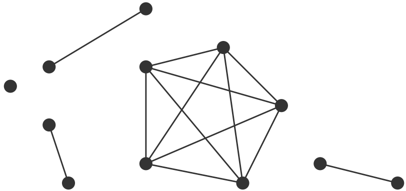
Degeneracy

- Degeneracy is closed under subgraphs.

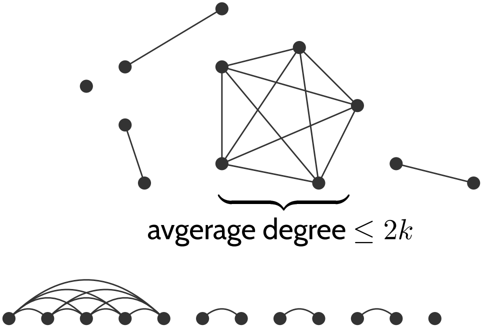


Degeneracy

- Degeneracy is closed under subgraphs.

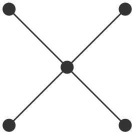


- Degeneracy is closed under subgraphs.
- If degeneracy is k then all subgraphs have average degree at most $2k$.

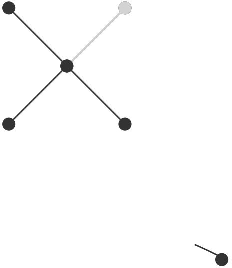


Degeneracy

An optimal degeneracy ordering can be computed by iteratively removing the vertex of smallest degree.

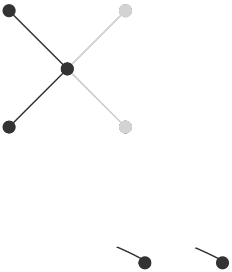


An optimal degeneracy ordering can be computed by iteratively removing the vertex of smallest degree.



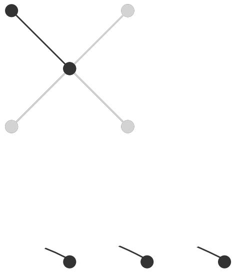
Degeneracy

An optimal degeneracy ordering can be computed by iteratively removing the vertex of smallest degree.



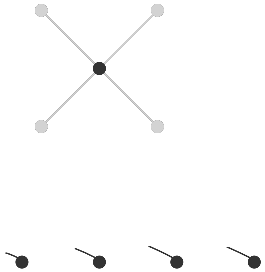
Degeneracy

An optimal degeneracy ordering can be computed by iteratively removing the vertex of smallest degree.



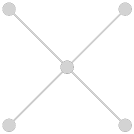
Degeneracy

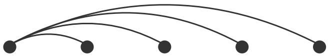
An optimal degeneracy ordering can be computed by iteratively removing the vertex of smallest degree.



Degeneracy

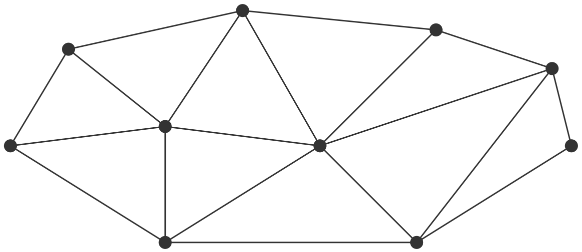
An optimal degeneracy ordering can be computed by iteratively removing the vertex of smallest degree.



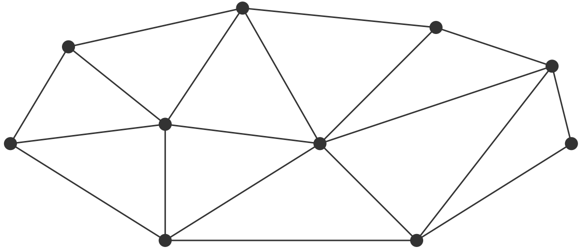


Degeneracy

Can we bound the degeneracy of a planar graph?



Can we bound the degeneracy of a planar graph? Every planar graph has vertex of degree ≤ 5. Iteratively pull this vertex out. Degeneracy is ≤ 5.



Degeneracy

Degeneracy is Too General

One cannot decide for graphs G of degeneracy 2 and FO sentences φ in time $f(|\varphi|)n^c$ whether $G \models \varphi$.

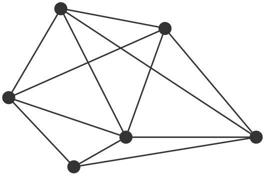
Degeneracy

Degeneracy is Too General

One cannot decide for graphs G of degeneracy 2 and FO sentences φ in time $f(|\varphi|)n^c$ whether $G \models \varphi$.

Reduction:

- We want to find a k-clique in graph G.



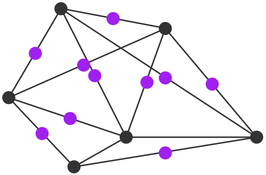
Degeneracy

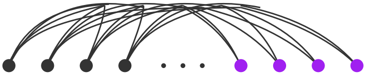
Degeneracy is Too General

One cannot decide for graphs G of degeneracy 2 and FO sentences φ in time $f(|\varphi|)n^c$ whether $G \models \varphi$.

Reduction:

- We want to find a k-clique in graph G.
- A subdivision G' of G has degeneracy ≤ 2.



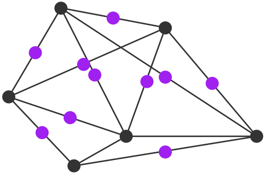


Degeneracy

Degeneracy is Too General

One cannot decide for graphs G of degeneracy 2 and FO sentences φ in time $f(|\varphi|)n^c$ whether $G \models \varphi$.

Reduction:



- We want to find a k-clique in graph G.
- A subdivision G' of G has degeneracy ≤ 2.
- If we could evaluate in time $f(k)|G'|^c$

$$G' \models \exists x_1 \text{ black}(x_1) \ldots x_k \text{ black}(x_k) \bigwedge_{i,j} \exists y \text{ purple}(y) \wedge x_i \sim y \wedge y \sim x_j$$

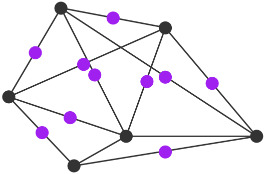
then we knew whether G had a clique of size k.

Degeneracy

Degeneracy is Too General

One cannot decide for graphs G of degeneracy 2 and FO sentences φ in time $f(|\varphi|)n^c$ whether $G \models \varphi$.

Reduction:



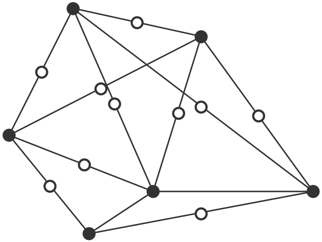
- We want to find a k-clique in graph G.
- A subdivision G' of G has degeneracy ≤ 2.
- If we could evaluate in time $f(k)|G'|^c$

  $$G' \models \exists x_1 \text{ black}(x_1) \ldots x_k \text{ black}(x_k) \bigwedge_{i,j} \exists y \text{ purple}(y) \wedge x_i \sim y \wedge y \sim x_j$$

  then we knew whether G had a clique of size k.
- But clique cannot be solved in FPT time.

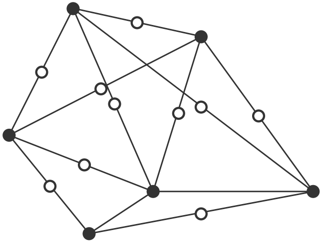
Sparsity Choices

- Do we consider subdivisions of arbitrary graphs "sparse"?



Sparsity Choices

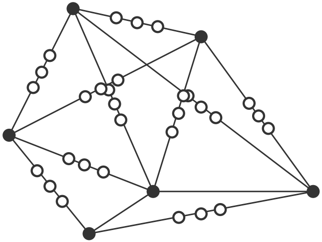
- Do we consider subdivisions of arbitrary graphs “sparse”?



- No, because FO-logic can recover the underlying graph at “depth” one

Sparsity Choices

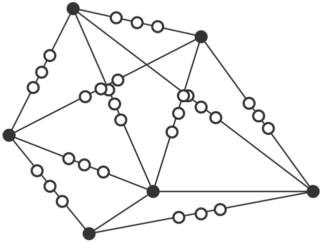
- Do we consider subdivisions of arbitrary graphs “sparse”?



- No, because FO-logic can recover the underlying graph at “depth” one and higher “depths.”

Sparsity Choices

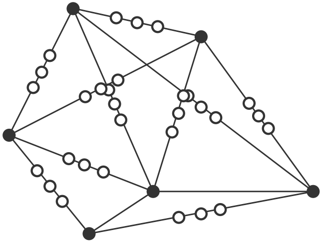
- Do we consider subdivisions of arbitrary graphs “sparse”?



- No, because FO-logic can recover the underlying graph at “depth” one and higher “depths.”
- We require that subgraphs at all “depths” are sparse.

Sparsity Choices

- Do we consider subdivisions of arbitrary graphs “sparse”?

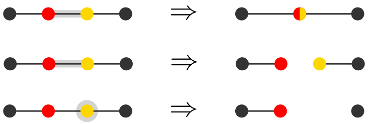


- No, because FO-logic can recover the underlying graph at “depth” one and higher “depths.”
- We require that subgraphs at all “depths” are sparse.
- Subgraphs hidden at a certain depth are called *shallow minors*.

Minors

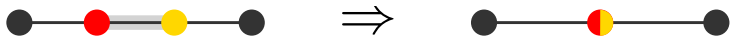
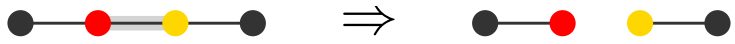
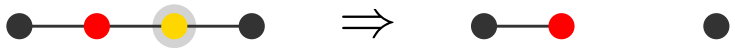
A graph H is a minor of G ($H \preccurlyeq G$) if H can be built from G by

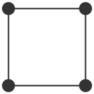
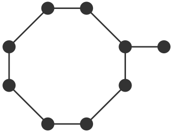
- merging edges
- removing edges
- removing vertices



Minors

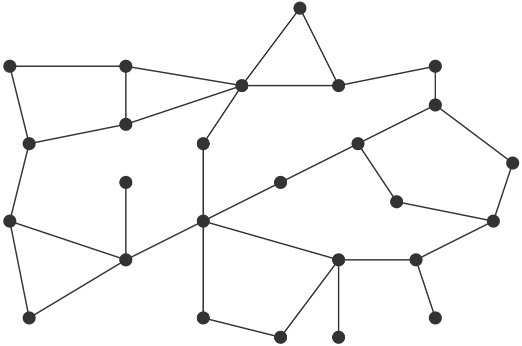
A graph H is a minor of G ($H \preccurlyeq G$) if H can be built from G by

- merging edges 
- removing edges 
- removing vertices 

 is minor of 

Minors

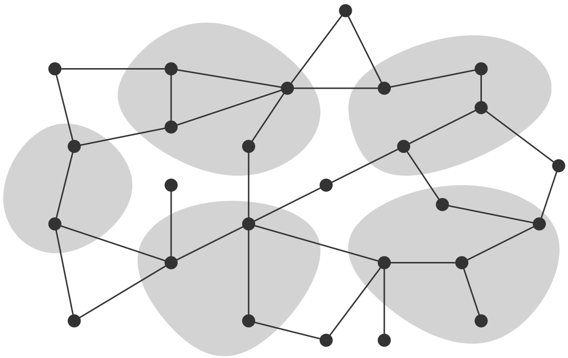
Equivalently, minors are graphs that are built from the host-graph by



Minors

Equivalently, minors are graphs that are built from the host-graph by

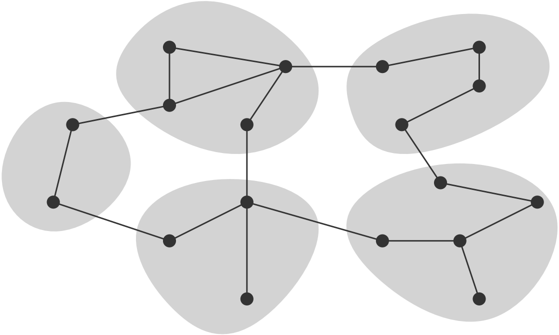
- picking some connected subgraphs



Minors

Equivalently, minors are graphs that are built from the host-graph by

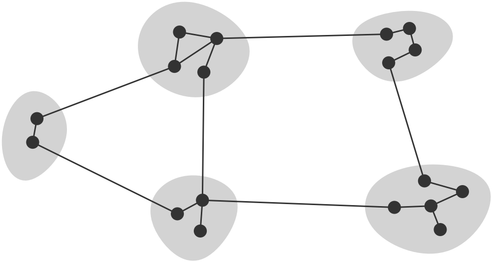
- picking some connected subgraphs
- removing all vertices outside these subgraphs



Minors

Equivalently, minors are graphs that are built from the host-graph by

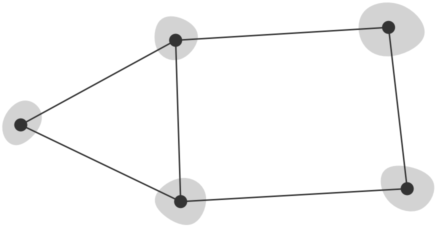
- picking some connected subgraphs
- removing all vertices outside these subgraphs
- merging each subgraph into a single vertex



Minors

Equivalently, minors are graphs that are built from the host-graph by

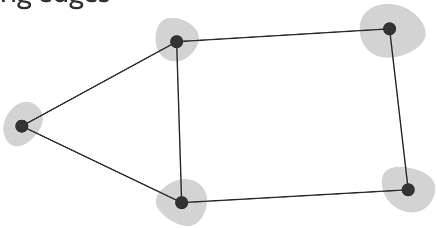
- picking some connected subgraphs
- removing all vertices outside these subgraphs
- merging each subgraph into a single vertex



Minors

Equivalently, minors are graphs that are built from the host-graph by

- picking some connected subgraphs
- removing all vertices outside these subgraphs
- merging each subgraph into a single vertex
- removing edges



Minor Characterization of Planar Graphs

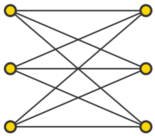
One can characterize graph classes by excluding certain minors.

Minor Characterization of Planar Graphs

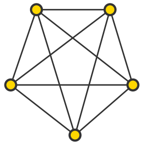
One can characterize graph classes by excluding certain minors.

Kuratowski's Theorem

A graph is planar iff it does not have $K_{3,3}$ or K_5 as a minor.



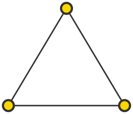
$K_{3,3}$



K_5

Minor Characterization of Treewidth

A graph is a forest iff it does not have K_3 as a minor.

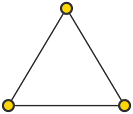


K_3

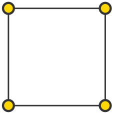
Minor Characterization of Treewidth

A graph is a forest iff it does not have K_3 as a minor.

A graph has treewidth ≤ 2 if it does not have K_4 as a minor.



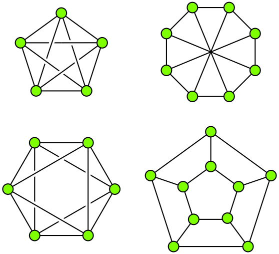
K_3



K_4

Minor Characterization of Treewidth

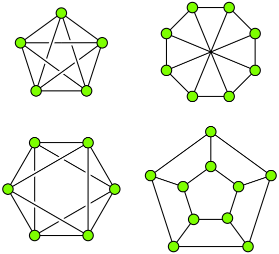
A graph has treewidth ≤ 3 iff it does not have any of these graphs as a minor.



Minor Characterization of Treewidth

A graph has treewidth ≤ 3 iff it does not have any of these graphs as a minor.

In general, for every k there exists a family of minors $\mathcal{F}_k$ such that a graph has treewidth $\leq k$ iff it does not have any graph from $\mathcal{F}_k$ as a minor.



Minor Characterization

Minor-Free Graphs

A class of graphs $\mathcal{C}$ is *minor-free* if there exists a t such that every graph in $\mathcal{C}$ excludes K_t as a minor.

Minor Characterization

Minor-Free Graphs

A class of graphs $\mathcal{C}$ is *minor-free* if there exists a t such that every graph in $\mathcal{C}$ excludes K_t as a minor.

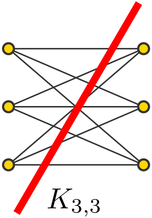
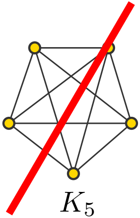
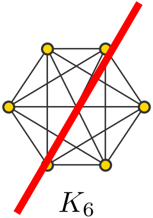
Every planar graph class and every class with bounded treewidth is minor-free.

Minor Characterization

Minor-Free Graphs

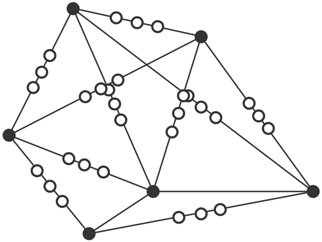
A class of graphs $\mathcal{C}$ is *minor-free* if there exists a t such that every graph in $\mathcal{C}$ excludes K_t as a minor.

Every planar graph class and every class with bounded treewidth is minor-free. Proof for planar graphs:

planar $\Rightarrow$  or  $\Rightarrow$ 

Sparsity

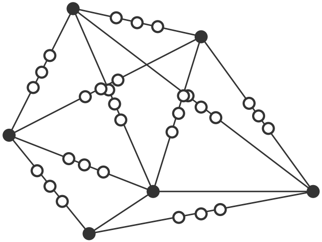
FO-logic can recover the underlying graph hidden at a “depth” in the graph. We require that subgraphs at all “depths” are sparse.



Can we say a graph is sparse if it is *minor-free*, i.e., does not contain any complicated minors?

Sparsity

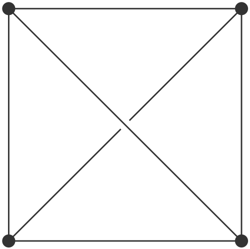
FO-logic can recover the underlying graph hidden at a “depth” in the graph. We require that subgraphs at all “depths” are sparse.



Can we say a graph is sparse if it is *minor-free*, i.e., does not contain any complicated minors? No. We could not capture locally bounded treewidth this way.

Subdivisions of High Depth

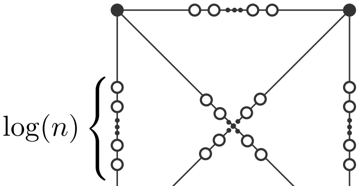
Remember the class of $\log(n)$-subdivisions of n-cliques.



clique of size n

Subdivisions of High Depth

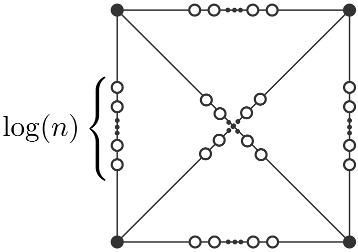
Remember the class of $\log(n)$-subdivisions of n-cliques.



clique of size n

Subdivisions of High Depth

Remember the class of $\log(n)$-subdivisions of n-cliques.

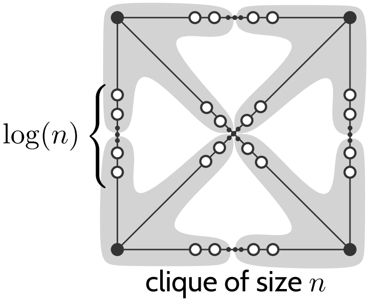


clique of size n

- We showed, it has locally bounded treewidth, and therefore fpt model-checking.

Subdivisions of High Depth

Remember the class of $\log(n)$-subdivisions of n-cliques.



- We showed, it has locally bounded treewidth, and therefore fpt model-checking.
- But it has arbitrary large cliques as minors.

Our Goal

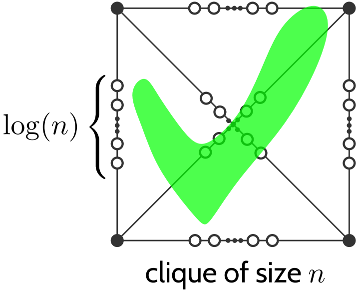
We want to say a graph class is “sparse” if

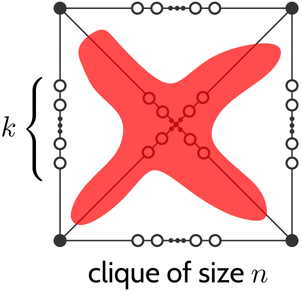
- it is closed under subgraphs,
- you can do FPT first-order model-checking on it.

Our Goal

We want to say a graph class is “sparse” if

- it is closed under subgraphs,
- you can do FPT first-order model-checking on it.

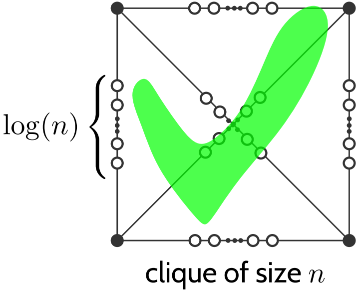


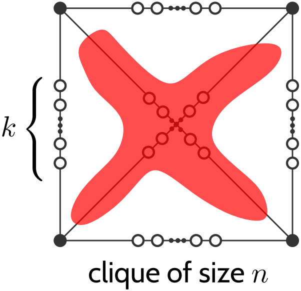


Our Goal

We want to say a graph class is “sparse” if

- it is closed under subgraphs,
- you can do FPT first-order model-checking on it.



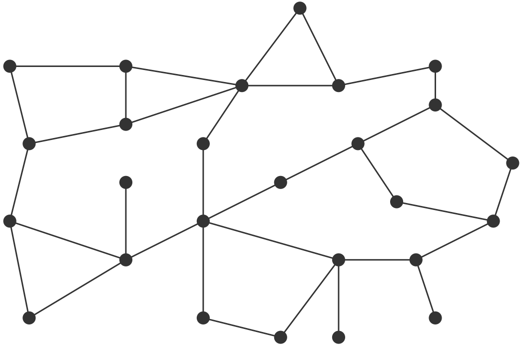


Dense minors are okay as long as they are at a high “depth”.

How do we formalize this?

Shallow Minors

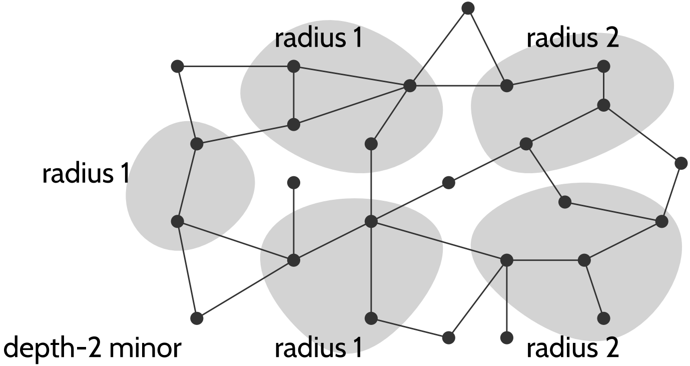
H is an *depth-r minor* of G ($H \preccurlyeq_r G$) if H can be built from G by



Shallow Minors

H is an *depth-r minor* of G ($H \preccurlyeq_r G$) if H can be built from G by

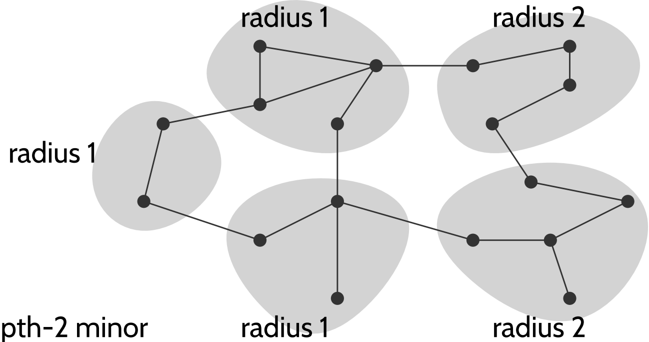
- picking some connected subgraphs with radius $\leq r$.



Shallow Minors

H is an *depth-r minor* of G ($H \preccurlyeq_r G$) if H can be built from G by

- picking some connected subgraphs with radius $\leq r$.
- removing all vertices outide these subgraphs

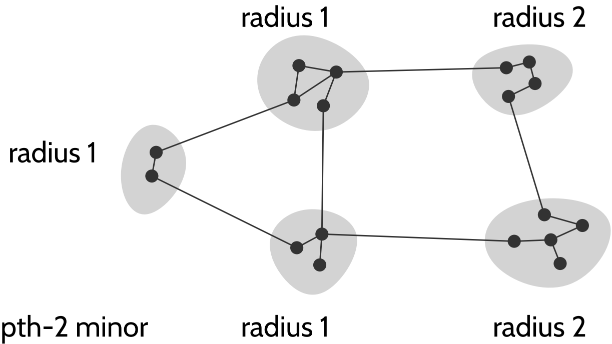


depth-2 minor

Shallow Minors

H is an *depth-r minor* of G ($H \preccurlyeq_r G$) if H can be built from G by

- picking some connected subgraphs with radius $\leq r$.
- removing all vertices outide these subgraphs
- merging each subgraph into a single vertex

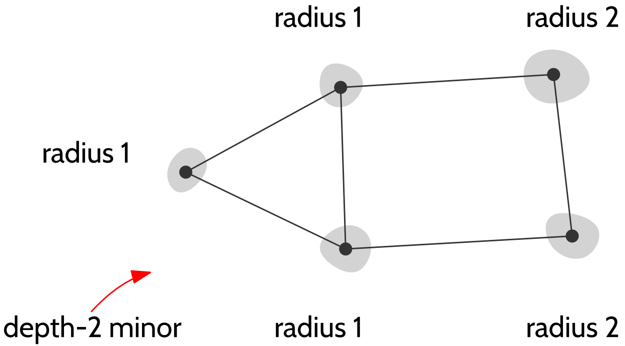


depth-2 minor

Shallow Minors

H is an *depth-r minor* of G ($H \preccurlyeq_r G$) if H can be built from G by

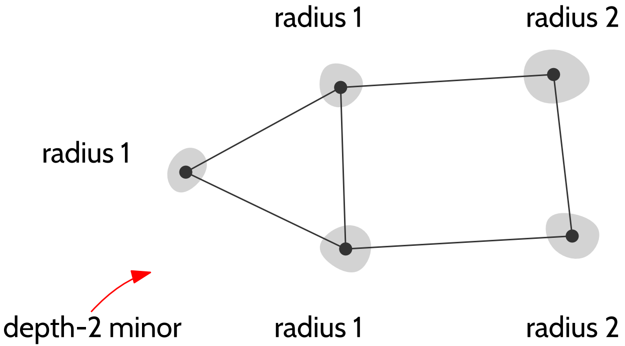
- picking some connected subgraphs with radius $\leq r$.
- removing all vertices outide these subgraphs
- merging each subgraph into a single vertex



Shallow Minors

H is an *depth-r minor* of G ($H \preccurlyeq_r G$) if H can be built from G by

- picking some connected subgraphs with radius $\leq r$.
- removing all vertices outide these subgraphs
- merging each subgraph into a single vertex
- removing edges



The Right Notion of Sparsity

We measure sparsity at depth r by measuring the depth-r minors of a graph G. This notion of sparsity was introduced by Nešetřil and Ossona de Mendez. We can think of two ways to do so

The Right Notion of Sparsity

We measure sparsity at depth r by measuring the depth-r minors of a graph G. This notion of sparsity was introduced by Nešetřil and Ossona de Mendez. We can think of two ways to do so

- bounding the average degree

$$\nabla_r(G) = \max\{\frac{|E(H)|}{|V(H)|} \mid H \preccurlyeq_r G\}$$

The Right Notion of Sparsity

We measure sparsity at depth r by measuring the depth-r minors of a graph G. This notion of sparsity was introduced by Nešetřil and Ossona de Mendez. We can think of two ways to do so

- bounding the average degree

$$\nabla_r(G) = \max\{\frac{|E(H)|}{|V(H)|} \mid H \preccurlyeq_r G\}$$

- bounding the clique size

$$\omega_r(G) = \max\{t \mid K_t \preccurlyeq_r G\}$$

The Right Notion of Sparsity

We measure sparsity at depth r by measuring the depth-r minors of a graph G. This notion of sparsity was introduced by Nešetřil and Ossona de Mendez. We can think of two ways to do so

- bounding the average degree

$$\nabla_r(G) = \max\{\frac{|E(H)|}{|V(H)|} \mid H \preccurlyeq_r G\}$$

- bounding the clique size

$$\omega_r(G) = \max\{t \mid K_t \preccurlyeq_r G\}$$

Bounded Expansion

A graph class $\mathcal{C}$ has bounded expansion if there exists a function $f(r)$ such that for all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.

The Right Notion of Sparsity

We measure sparsity at depth r by measuring the depth-r minors of a graph G. This notion of sparsity was introduced by Nešetřil and Ossona de Mendez. We can think of two ways to do so

- bounding the average degree

$$\nabla_r(G) = \max\left\{\frac{|E(H)|}{|V(H)|} \mid H \preccurlyeq_r G\right\}$$

- bounding the clique size

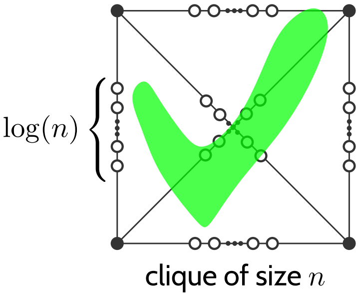
$$\omega_r(G) = \max\{t \mid K_t \preccurlyeq_r G\}$$

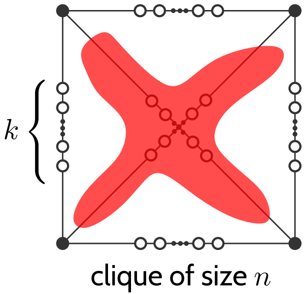
Nowhere Dense

A graph class $\mathcal{C}$ is nowhere dense if there exists a function $f(r)$ such that for all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\omega_r(G) \leq f(r)$.

Which of these classes has bounded expansion / is nowhere dense?

Which of these classes has bounded expansion / is nowhere dense?





Our Goal

We want to say a graph class is "sparse" if

- it is closed under subgraphs,
- you can do FPT first-order model-checking on it.

Our Goal

We want to say a graph class is “sparse” if

- it is closed under subgraphs,
- you can do FPT first-order model-checking on it.

Theorem (Grohe, Kreuzer, Siebertz 2017)

For graph class $\mathcal{C}$ that is closed under subgraphs holds $\mathcal{C}$ is nowhere dense iff the first-order model-checking problem on $\mathcal{C}$ is FPT (assuming FPT $\neq$ AW[$*$]).

Our Goal

We want to say a graph class is "sparse" if

- it is closed under subgraphs,
- you can do FPT first-order model-checking on it.

Theorem (Grohe, Kreuzer, Siebertz 2017)

For graph class $\mathcal{C}$ that is closed under subgraphs holds $\mathcal{C}$ is nowhere dense iff the first-order model-checking problem on $\mathcal{C}$ is FPT (assuming FPT $\neq$ AW[$*$]).

Before we prove (parts of) this result in the next lectures, we discuss the relationship to other graph classes.

Many Sparse Graph Classes

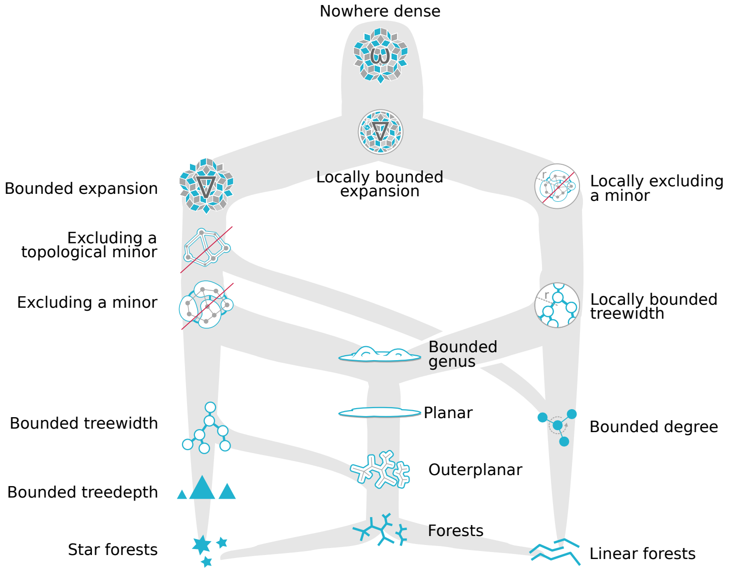


Figure by Felix Reidl

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$:

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$: If $\omega_r(G)$ is large then also $\nabla_r(G)$:

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$: If $\omega_r(G)$ is large then also $\nabla_r(G)$:

$$\omega_r(G) = t \;\; \Rightarrow$$

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$: If $\omega_r(G)$ is large then also $\nabla_r(G)$:

$$\omega_r(G) = t \;\Rightarrow\; \nabla_r(G) \geq \frac{|E(K_t)|}{|V(K_t)|}$$

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$: If $\omega_r(G)$ is large then also $\nabla_r(G)$:

$$\omega_r(G) = t \;\Rightarrow\; \nabla_r(G) \geq \frac{|E(K_t)|}{|V(K_t)|} = \frac{t(t-1)/2}{t} = \frac{t-1}{2}$$

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$: If $\omega_r(G)$ is large then also $\nabla_r(G)$:

$$\omega_r(G) = t \;\Rightarrow\; \nabla_r(G) \geq \frac{|E(K_t)|}{|V(K_t)|} = \frac{t(t-1)/2}{t} = \frac{t-1}{2}$$

- for all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\omega_r(G) \leq 2f(r) + 1$.

Nowhere Dense vs. Bounded Expansion

Every graph class with bounded expansion is also nowhere dense.

Proof:

- Assume $\mathcal{C}$ has bounded expansion with function $f(r)$.
- For all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\nabla_r(G) \leq f(r)$.
- $\omega_r(G) \leq 2\nabla_r(G) + 1$: If $\omega_r(G)$ is large then also $\nabla_r(G)$:

$$\omega_r(G) = t \;\Rightarrow\; \nabla_r(G) \geq \frac{|E(K_t)|}{|V(K_t)|} = \frac{t(t-1)/2}{t} = \frac{t-1}{2}$$

- for all $r \in \mathbb{N}$ and all $G \in \mathcal{C}$ we have $\omega_r(G) \leq 2f(r) + 1$.
- So $\mathcal{C}$ is nowhere dense with function $2f(r) + 1$.

Nowhere Dense vs. Bounded Expansion

But there exist graph classes that are nowhere dense but do not have bounded expansion.

Nowhere Dense vs. Bounded Expansion

But there exist graph classes that are nowhere dense but do not have bounded expansion.

The few known such graph classes have high average degree and high girth.

Nowhere Dense vs. Bounded Expansion

> But there exist graph classes that are nowhere dense but do not have bounded expansion.

The few known such graph classes have high average degree and high girth. For example random graphs with edge probability $\log(n)/n$.

Proof:

Nowhere Dense vs. Bounded Expansion

But there exist graph classes that are nowhere dense but do not have bounded expansion.

The few known such graph classes have high average degree and high girth. For example random graphs with edge probability $\log(n)/n$.

Proof:

- The expected degree is $2\log(n)$. So with high probably unbounded average degree (already at depth zero) and therefore not bounded expansion.

Nowhere Dense vs. Bounded Expansion

But there exist graph classes that are nowhere dense but do not have bounded expansion.

The few known such graph classes have high average degree and high girth. For example random graphs with edge probability $\log(n)/n$.

Proof:

- The expected degree is $2\log(n)$. So with high probably unbounded average degree (already at depth zero) and therefore not bounded expansion.
- Probability that vertices $v_1, \ldots, v_k$ form a cycle is $\left(\log(n)/n\right)^k$.

Nowhere Dense vs. Bounded Expansion

> But there exist graph classes that are nowhere dense but do not have bounded expansion.

The few known such graph classes have high average degree and high girth. For example random graphs with edge probability $\log(n)/n$.

Proof:

- The expected degree is $2\log(n)$. So with high probably unbounded average degree (already at depth zero) and therefore not bounded expansion.
- Probability that vertices $v_1, \ldots, v_k$ form a cycle is $\left(\log(n)/n\right)^k$.
- Expected number of k-cycles is $n^k \cdot \left(\log(n)/n\right)^k = \log(n)^k$.

Nowhere Dense vs. Bounded Expansion

- Remove all cycles of length $\leq \log(\log(n))$. The number of them is roughly $\log(n)^{\log(\log(n))}$. This is not too much.

Nowhere Dense vs. Bounded Expansion

- Remove all cycles of length $\leq \log(\log(n))$. The number of them is roughly $\log(n)^{\log(\log(n))}$. This is not too much.
- Only a vanishing fraction of vertices has been removed. The expected degree is still roughly $2\log(n)$. So with high probably still not bounded expansion.

Nowhere Dense vs. Bounded Expansion

- Remove all cycles of length $\leq \log(\log(n))$. The number of them is roughly $\log(n)^{\log(\log(n))}$. This is not too much.
- Only a vanishing fraction of vertices has been removed. The expected degree is still roughly $2\log(n)$. So with high probably still not bounded expansion.
- However, every $\log(\log(n))$-neighborhood is a tree.

Nowhere Dense vs. Bounded Expansion

- Remove all cycles of length $\leq \log(\log(n))$. The number of them is roughly $\log(n)^{\log(\log(n))}$. This is not too much.
- Only a vanishing fraction of vertices has been removed. The expected degree is still roughly $2\log(n)$. So with high probably still not bounded expansion.
- However, every $\log(\log(n))$-neighborhood is a tree.
- This means every $\log(\log(n))/6$-shallow minor is triangle-free and therefore $\omega_r(G) \leq 2$ for $r \leq \log(\log(n))/6$.

Nowhere Dense vs. Bounded Expansion

- Remove all cycles of length $\leq \log(\log(n))$. The number of them is roughly $\log(n)^{\log(\log(n))}$. This is not too much.
- Only a vanishing fraction of vertices has been removed. The expected degree is still roughly $2\log(n)$. So with high probably still not bounded expansion.
- However, every $\log(\log(n))$-neighborhood is a tree.
- This means every $\log(\log(n))/6$-shallow minor is triangle-free and therefore $\omega_r(G) \leq 2$ for $r \leq \log(\log(n))/6$.
- On the other hand, if $r \geq \log(\log(n))/6$ then $\omega_r(G) \leq n \leq 2^{2^{6r}}$.

Nowhere Dense vs. Bounded Expansion

- Remove all cycles of length $\leq \log(\log(n))$. The number of them is roughly $\log(n)^{\log(\log(n))}$. This is not too much.
- Only a vanishing fraction of vertices has been removed. The expected degree is still roughly $2\log(n)$. So with high probably still not bounded expansion.
- However, every $\log(\log(n))$-neighborhood is a tree.
- This means every $\log(\log(n))/6$-shallow minor is triangle-free and therefore $\omega_r(G) \leq 2$ for $r \leq \log(\log(n))/6$.
- On the other hand, if $r \geq \log(\log(n))/6$ then $\omega_r(G) \leq n \leq 2^{2^{6r}}$.
- Thus the graph comes with high probably from a nowhere dense class.

Many Sparse Graph Classes

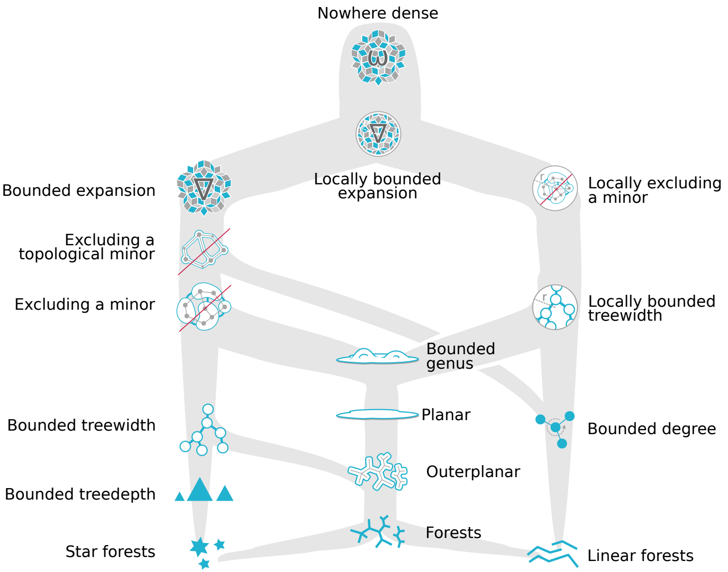


Figure by Felix Reidl

Bounded Expansion vs. Bounded Degree

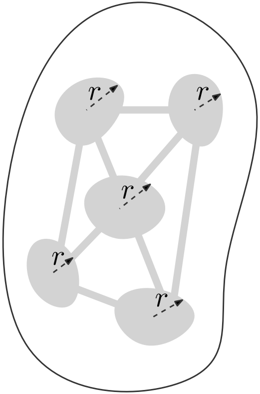
Every graph class with bounded degree has bounded expansion. In particular, if a graph G has degree at most d then $\nabla_r(G) \leq d^{r+2}$.

Bounded Expansion vs. Bounded Degree

Every graph class with bounded degree has bounded expansion. In particular, if a graph G has degree at most d then $\nabla_r(G) \leq d^{r+2}$.

Proof:

- Depth r-minors are built from subgraphs with radius $\leq r$.

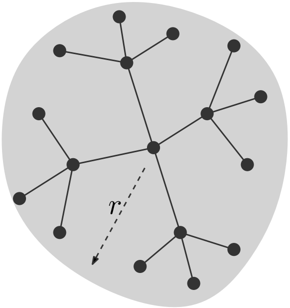


Bounded Expansion vs. Bounded Degree

Every graph class with bounded degree has bounded expansion. In particular, if a graph G has degree at most d then $\nabla_r(G) \leq d^{r+2}$.

Proof:

- Depth r-minors are built from subgraphs with radius $\leq r$.
- As proven earlier, these subgraphs have size at most d^{r+1}.

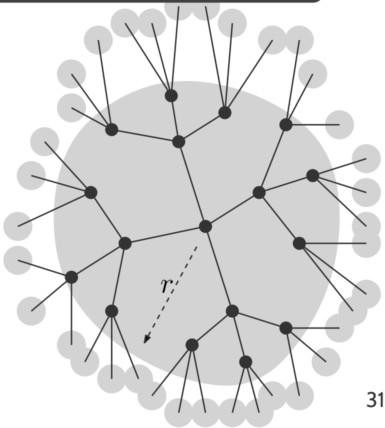


Bounded Expansion vs. Bounded Degree

Every graph class with bounded degree has bounded expansion. In particular, if a graph G has degree at most d then $\nabla_r(G) \leq d^{r+2}$.

Proof:

- Depth r-minors are built from subgraphs with radius $\leq r$.
- As proven earlier, these subgraphs have size at most d^{r+1}.
- Such a subgraph can have at most d^{r+2} neighbors in the minor model.

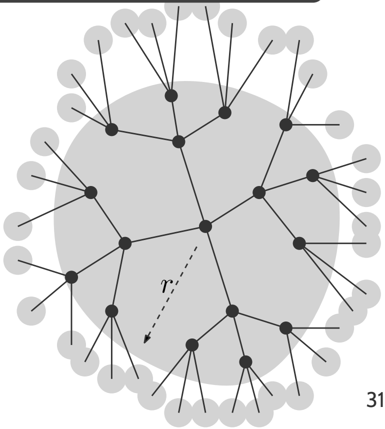


Bounded Expansion vs. Bounded Degree

Every graph class with bounded degree has bounded expansion. In particular, if a graph G has degree at most d then $\nabla_r(G) \leq d^{r+2}$.

Proof:

- Depth r-minors are built from subgraphs with radius $\leq r$.
- As proven earlier, these subgraphs have size at most d^{r+1}.
- Such a subgraph can have at most d^{r+2} neighbors in the minor model.
- The maximum degree of an depth r-minor is d^{r+2}.



Many Sparse Graph Classes

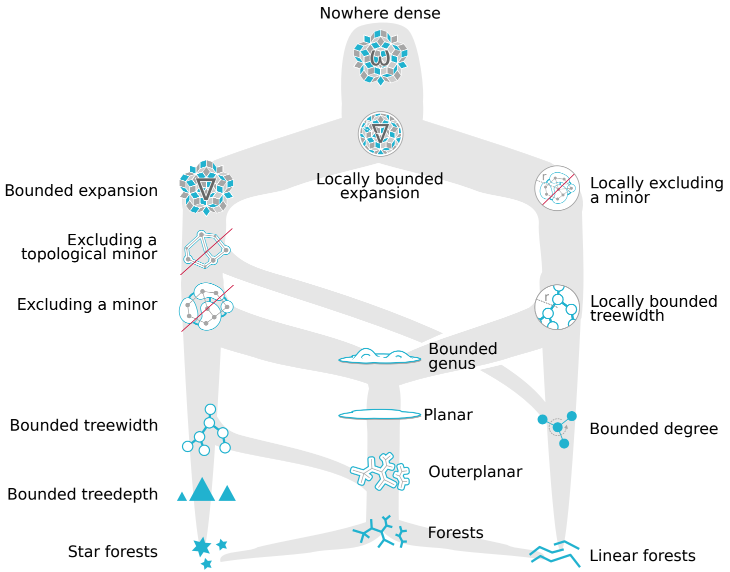


Figure by Felix Reidl

Bounded Expansion vs. Minor Free Classes

Let $\mathcal{C}$ be a minor-free graph class (examples are planar graphs, or bounded treewidth.) Then $\mathcal{C}$ has bounded expansion.

Bounded Expansion vs. Minor Free Classes

Let $\mathcal{C}$ be a minor-free graph class (examples are planar graphs, or bounded treewidth.) Then $\mathcal{C}$ has bounded expansion.

Proof:

- Because $\mathcal{C}$ is minor-free, it excludes some K_t as a minor.

Bounded Expansion vs. Minor Free Classes

Let $\mathcal{C}$ be a minor-free graph class (examples are planar graphs, or bounded treewidth.) Then $\mathcal{C}$ has bounded expansion.

Proof:

- Because $\mathcal{C}$ is minor-free, it excludes some K_t as a minor.
- Let $G \in \mathcal{C}$. Then G does not have K_t as a minor.

Bounded Expansion vs. Minor Free Classes

Let $\mathcal{C}$ be a minor-free graph class (examples are planar graphs, or bounded treewidth.) Then $\mathcal{C}$ has bounded expansion.

Proof:

- Because $\mathcal{C}$ is minor-free, it excludes some K_t as a minor.
- Let $G \in \mathcal{C}$. Then G does not have K_t as a minor.
- Also every r-shallow minor of G does not have K_t as a minor.

Bounded Expansion vs. Minor Free Classes

Let $\mathcal{C}$ be a minor-free graph class (examples are planar graphs, or bounded treewidth.) Then $\mathcal{C}$ has bounded expansion.

Proof:

- Because $\mathcal{C}$ is minor-free, it excludes some K_t as a minor.
- Let $G \in \mathcal{C}$. Then G does not have K_t as a minor.
- Also every r-shallow minor of G does not have K_t as a minor.
- Extra lemma: Every graph that does not contain K_t as a minor has $< 2^t$ edges per vertex.

Bounded Expansion vs. Minor Free Classes

Let $\mathcal{C}$ be a minor-free graph class (examples are planar graphs, or bounded treewidth.) Then $\mathcal{C}$ has bounded expansion.

Proof:

- Because $\mathcal{C}$ is minor-free, it excludes some K_t as a minor.
- Let $G \in \mathcal{C}$. Then G does not have K_t as a minor.
- Also every r-shallow minor of G does not have K_t as a minor.
- Extra lemma: Every graph that does not contain K_t as a minor has $< 2^t$ edges per vertex.
- This means every r-shallow minor of G has $< 2^t$ edges per vertex. In other words, $\Delta_r(G) \leq 2^t$ for all $r \in \mathbb{N}$.

Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

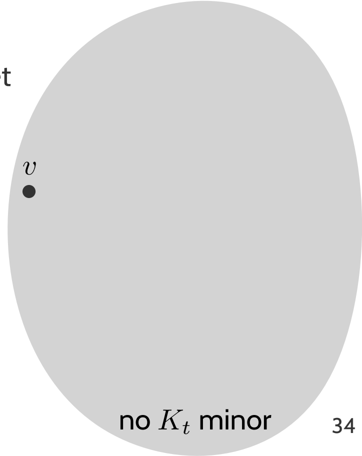
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.



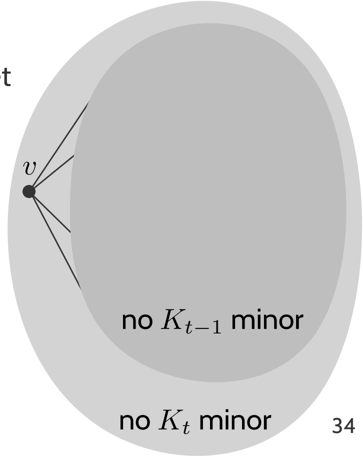
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor.



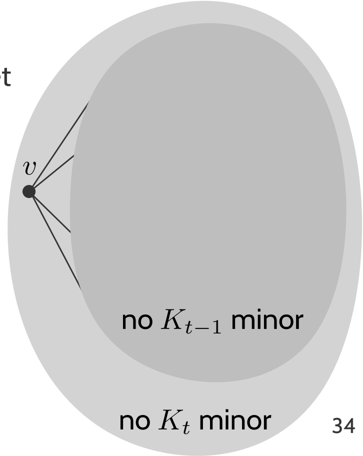
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor.
  By induction $< 2^{t-1}$ edges per vertex.



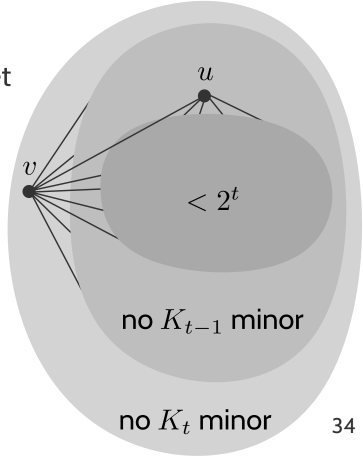
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor.
  By induction $< 2^{t-1}$ edges per vertex.
- We can pick u of degree $< 2^t$ in $G[N(v)]$.



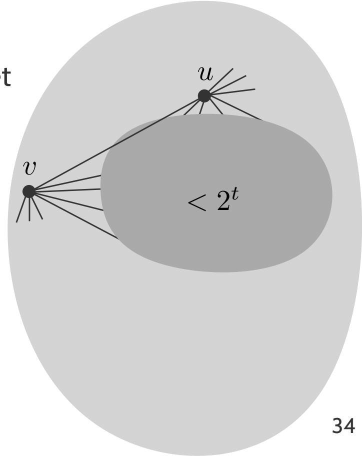
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor. By induction $< 2^{t-1}$ edges per vertex.
- We can pick u of degree $< 2^t$ in $G[N(v)]$.
- Contract u and v.



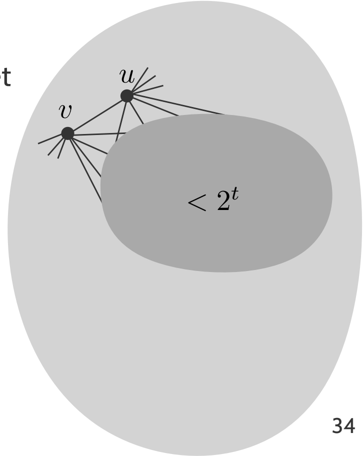
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor.
  By induction $< 2^{t-1}$ edges per vertex.
- We can pick u of degree $< 2^t$ in $G[N(v)]$.
- Contract u and v.



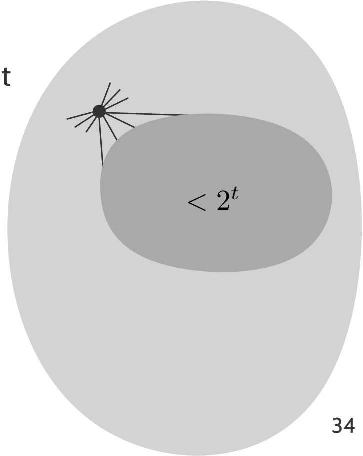
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor.
  By induction $< 2^{t-1}$ edges per vertex.
- We can pick u of degree $< 2^t$ in $G[N(v)]$.
- Contract u and v. The result has by induction $< 2^t$ edges per vertex.



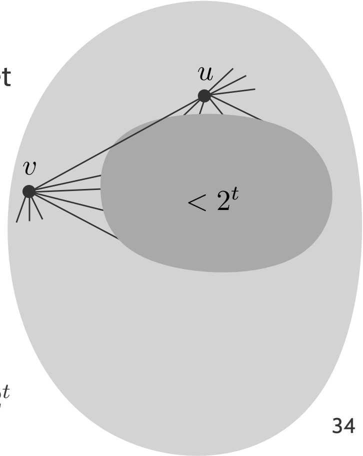
Bounded Expansion vs. Minor Free Classes

Lemma 1.16, Chapter 1, MIMUW Sparsity Lecture Notes

For every $t \geq 2$, if a graph does not contain K_t as a minor, then it has $< 2^t$ edges per vertex.

Proof (induction over graph size and t):

- Let G be a graph with no K_t minor and let $v \in V(G)$.
- Then $G[N(v)]$ has no K_{t-1} minor.
  By induction $< 2^{t-1}$ edges per vertex.
- We can pick u of degree $< 2^t$ in $G[N(v)]$.
- Contract u and v. The result has by induction $< 2^t$ edges per vertex.
- Going back to G adds one vertex and $< 2^t$ edges.



Many Sparse Graph Classes

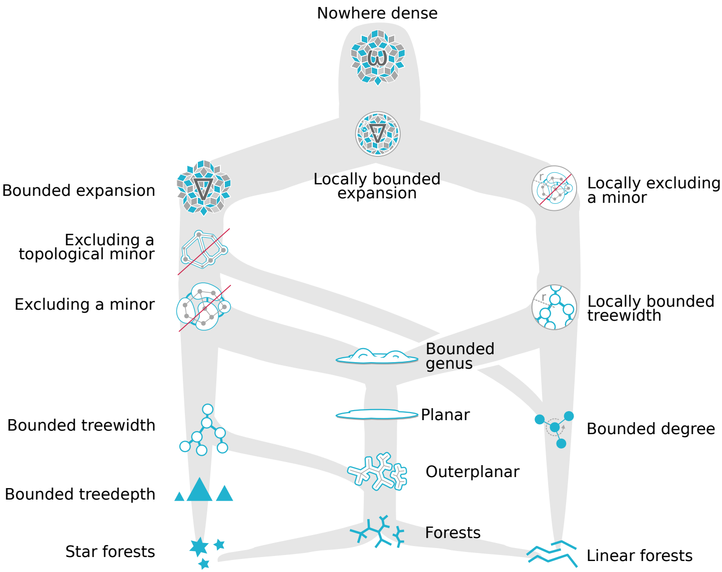


Figure by Felix Reidl

Main Results for Sparse Graphs

Theorem (Dvořák, Král, Thomas 2013)

Let $\mathcal{C}$ be a graph class with bounded expansion. There exists a function f such that for every FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(|\varphi|)n$.

Main Results for Sparse Graphs

Theorem (Dvořák, Král, Thomas 2013)

Let $\mathcal{C}$ be a graph class with bounded expansion. There exists a function f such that for every FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(|\varphi|)n$.

Theorem (Grohe, Kreuzer, Siebertz 2017)

Let $\mathcal{C}$ be a nowhere dense graph class. There exists a function f such that for every $\varepsilon > 0$, FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(\varepsilon, |\varphi|)n^{1+\varepsilon}$.

Main Results for Sparse Graphs

Theorem (Dvořák, Král, Thomas 2013)

Let $\mathcal{C}$ be a graph class with bounded expansion. There exists a function f such that for every FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(|\varphi|)n$.

Theorem (Grohe, Kreuzer, Siebertz 2017)

Let $\mathcal{C}$ be a nowhere dense graph class. There exists a function f such that for every $\varepsilon > 0$, FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(\varepsilon, |\varphi|)n^{1+\varepsilon}$.

General rule: Things that work for bounded expansion also work for nowhere dense, but in an uglier way. This is why we focus on bounded expansion only in this course.

Existential Model-Checking

We will first prove a weaker result that is a building block in many other algorithms.

Let $\mathcal{C}$ be a class with bounded expansion. There exists a function f such that for every *existential* FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(|\varphi|)n$.

Existential Model-Checking

We will first prove a weaker result that is a building block in many other algorithms.

> Let $\mathcal{C}$ be a class with bounded expansion. There exists a function f such that for every *existential* FO formula φ and graph $G \in \mathcal{C}$ one can decide whether $G \models \varphi$ in time $f(|\varphi|)n$.

This is more or less equivalent to the following.

> Let $\mathcal{C}$ be a class with bounded expansion. There exists a function f such that for every pattern graph H and host graph $G \in \mathcal{C}$ one can decide whether H is an induced subgraph of G in time $f(|H|)n$.

Existential Model-Checking

Proof of equivalence:

- Assume we want to know whether $G \models \varphi$ for some existential formula with q quantifiers. For example $\varphi = \exists x \exists y \exists z x{\sim}y \wedge y{\not\sim}z$.

Existential Model-Checking

Proof of equivalence:

- Assume we want to know whether $G \models \varphi$ for some existential formula with q quantifiers. For example $\varphi = \exists x \exists y \exists z x{\sim}y \wedge y{\not\sim}z$.
- Compute set $\mathcal{H}$ of all graphs with at most q vertices and $H \models \varphi$. In our case,

$\mathcal{H} = \{$  $\}$

Proof of equivalence:

- Assume we want to know whether $G \models \varphi$ for some existential formula with q quantifiers. For example $\varphi = \exists x \exists y \exists z x{\sim}y \wedge y{\not\sim}z$.
- Compute set $\mathcal{H}$ of all graphs with at most q vertices and $H \models \varphi$. In our case,

$\mathcal{H} = \{$  $\}$

- Now $G \models \varphi$ iff G contains some graph from $\mathcal{H}$ as induced subgraph.
  - Assume $G \models \varphi$. Then the satisfying assignment describes induced subgraph H of G with $H \models \varphi$.
  - Assume $H \in \mathcal{H}$ is induced subgraph of G. Then $H \models \varphi$. This does not change while adding the remaining vertices of G.

Alternative Characterizations

How can we prove these results?

- Gaifman does not help much because neighborhoods can be the whole graph.
- So far, all we know that certain shallow minors are not present.
- If we have a better understanding of the structure of sparse graphs, this will help us.

Alternative Characterizations

There are many alternative definitions of bounded expansion and nowhere dense classes.

- shallow minors
- generalized coloring numbers
- low treedepth colorings
- transitive fraternal augmentations
- quasi-wideness
- connector-splitter games

Which one is best depends on the task.

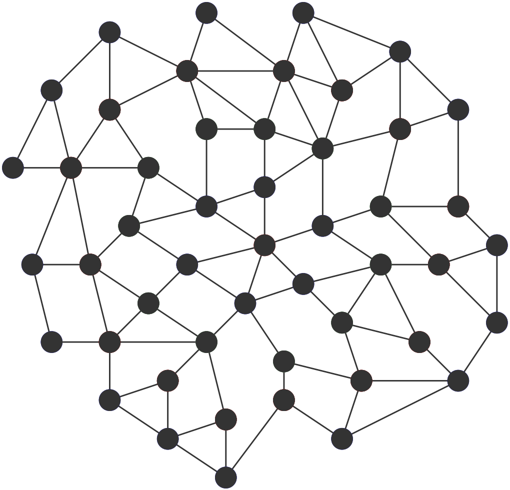
To prove the result, we will use the powerful notion of *low treedepth colorings*.

To prove the result, we will use the powerful notion of *low treedepth colorings*.

As a warmup, we solve the problem on planar graphs and then generalize the approach to bounded expansion.

Baker's Technique

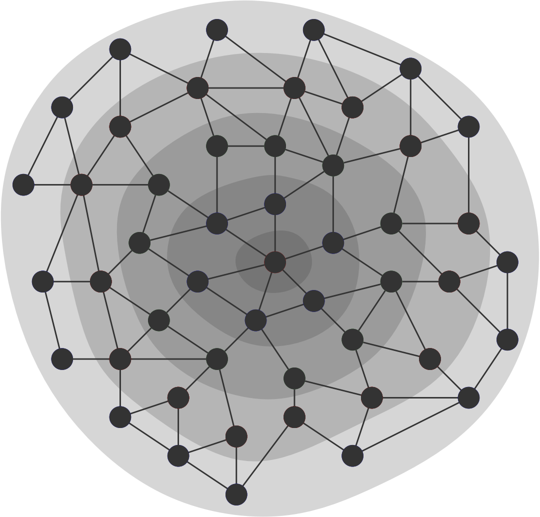
On any planar graph you can do the following.



Baker's Technique

On any planar graph you can do the following.

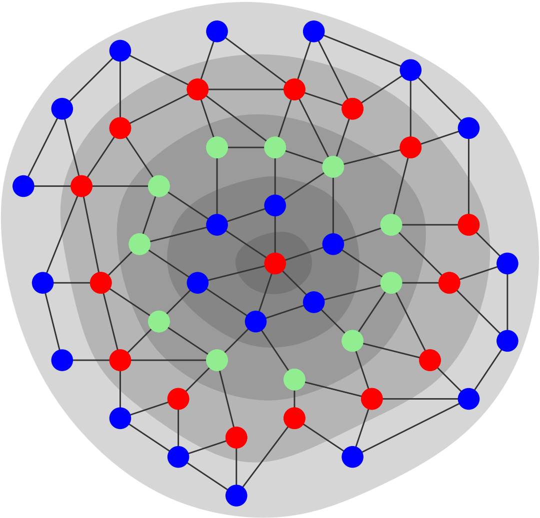
- do breadth-first search



Baker's Technique

On any planar graph you can do the following.

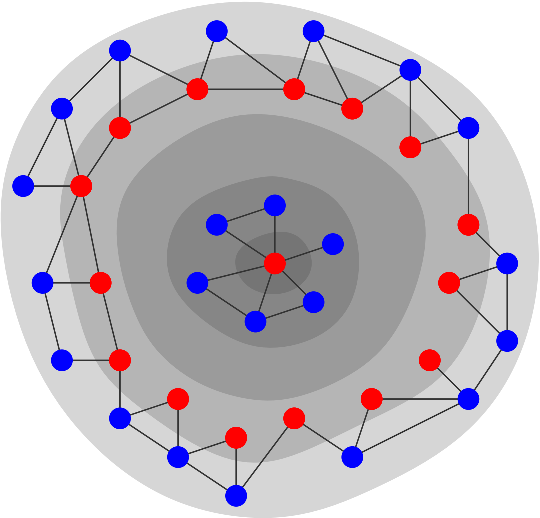
- do breadth-first search
- give layer i color $i \bmod p$



Baker's Technique

On any planar graph you can do the following.

- do breadth-first search
- give layer i color i mod p
- pick a strict subset of colors

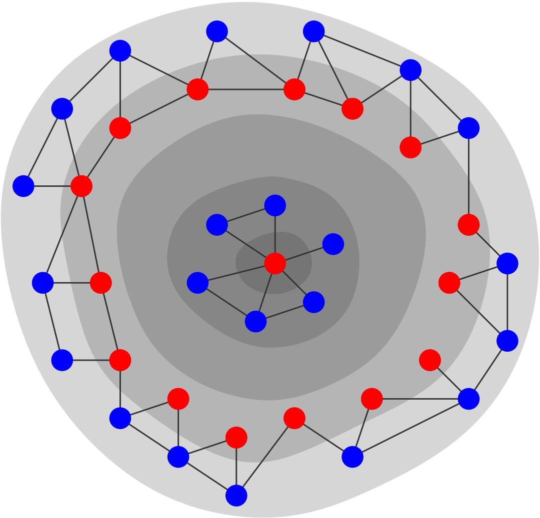


Baker's Technique

On any planar graph you can do the following.

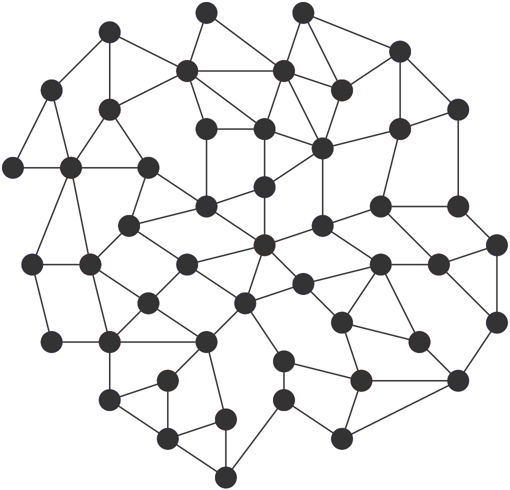
- do breadth-first search
- give layer i color i mod p
- pick a strict subset of colors

The resulting graph has treewidth at most $3p + 1$.



Baker's Technique

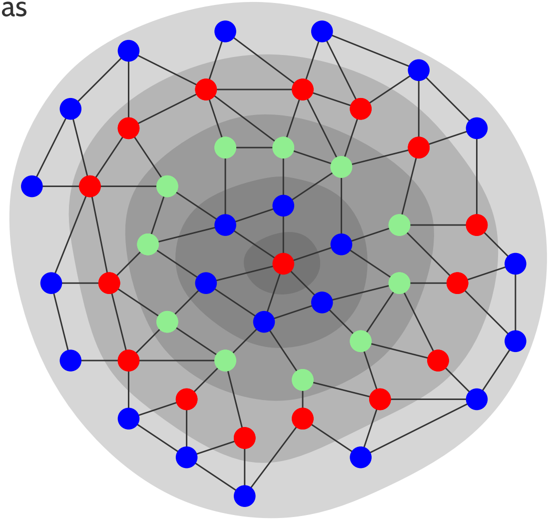
We want to know whether G has H as an induced subgraph.



Baker's Technique

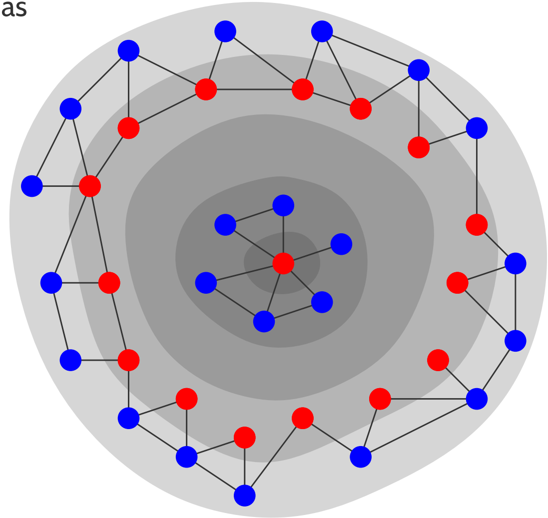
We want to know whether G has H as an induced subgraph.

- color the graph as before with $p = |H| + 1$ colors.



We want to know whether G has H as an induced subgraph.

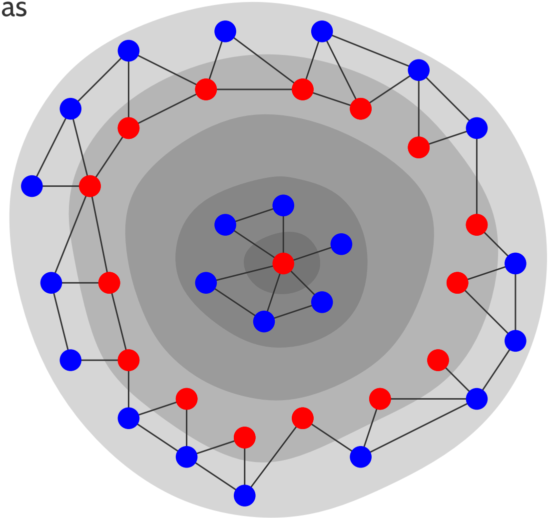
- color the graph as before with $p = |H| + 1$ colors.
- If H is induced subgraph then it is contained in a subset of $|H|$ colors.



Baker's Technique

We want to know whether G has H as an induced subgraph.

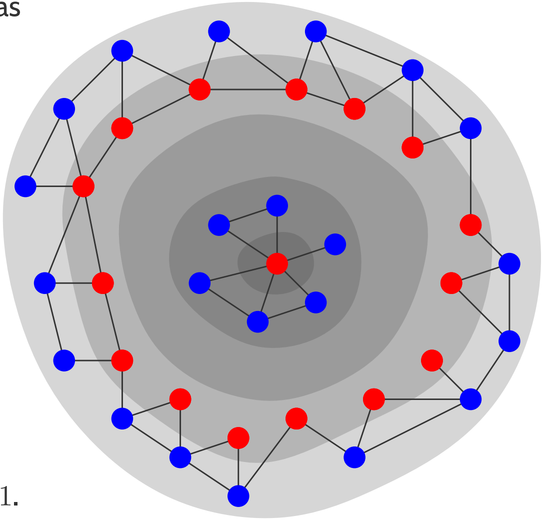
- color the graph as before with $p = |H| + 1$ colors.
- If H is induced subgraph then it is contained in a subset of $|H|$ colors.
- Enumerate all subsets of $|H|$ colors and search for H.



Baker's Technique

We want to know whether G has H as an induced subgraph.

- color the graph as before with $p = |H| + 1$ colors.
- If H is induced subgraph then it is contained in a subset of $|H|$ colors.
- Enumerate all subsets of $|H|$ colors and search for H.
- Use Courcelle on induced graph of treewith $\leq 3p + 1$.

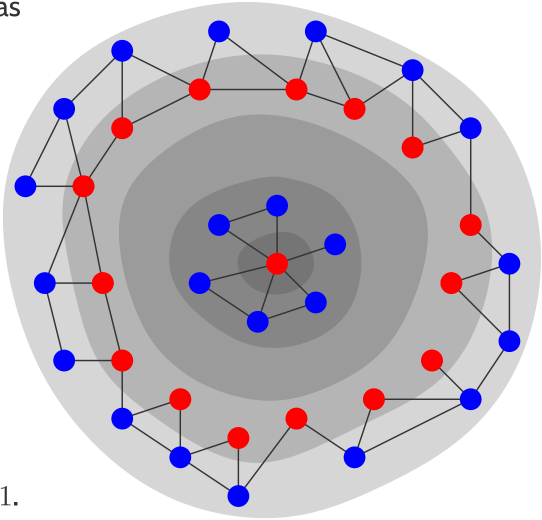


Baker's Technique

We want to know whether G has H as an induced subgraph.

- color the graph as before with $p = |H| + 1$ colors.
- If H is induced subgraph then it is contained in a subset of $|H|$ colors.
- Enumerate all subsets of $|H|$ colors and search for H.
- Use Courcelle on induced graph of treewith $\leq 3p + 1$.

Run time $p \cdot f(3p + 1, |H|) \cdot n$.



Low Treedepth Colorings

We used the following observation of planar graphs.

- For every p one can color the graph with $p + 1$ colors such that every set of p colors induces a graph with treewith at most $3p + 1$.

We used the following observation of planar graphs.

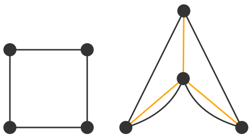
- For every p one can color the graph with $p+1$ colors such that every set of p colors induces a graph with treewith at most $3p+1$.

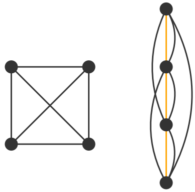
We can get something similar for bounded expansion.

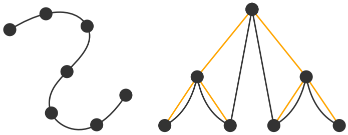
- For every p one can color the graph with $f(p)$ colors such that every set of p colors induces a graph with treedepth at most p.

Definition

The *treedepth* of a graph G is the minimum height of a rooted forest F such that all edges of G go between ancestors and descendants in F.







Treedepth vs. Treewidth

> The treewidth of a graph is as most as large as the treedepth.

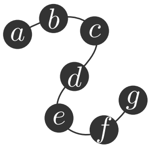
This means graph classes with bounded treewidth are more general than those with bounded treedepth.

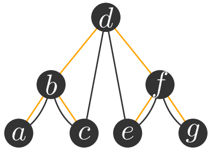
Treedepth vs. Treewidth

The treewidth of a graph is as most as large as the treedepth.

This means graph classes with bounded treewidth are more general than those with bounded treedepth.

Proof: The set of all paths from leafs to the root yield a tree decomposition.



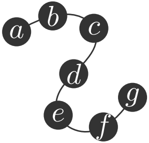


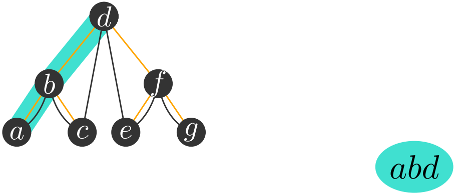
Treedepth vs. Treewidth

The treewidth of a graph is as most as large as the treedepth.

This means graph classes with bounded treewidth are more general than those with bounded treedepth.

Proof: The set of all paths from leafs to the root yield a tree decomposition.



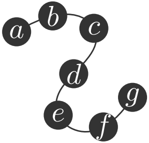


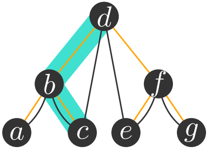
Treedepth vs. Treewidth

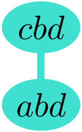
The treewidth of a graph is as most as large as the treedepth.

This means graph classes with bounded treewidth are more general than those with bounded treedepth.

Proof: The set of all paths from leafs to the root yield a tree decomposition.





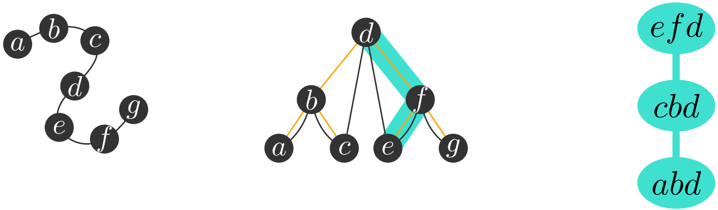


Treedepth vs. Treewidth

The treewidth of a graph is as most as large as the treedepth.

This means graph classes with bounded treewidth are more general than those with bounded treedepth.

Proof: The set of all paths from leafs to the root yield a tree decomposition.

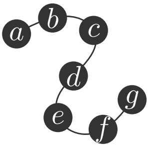


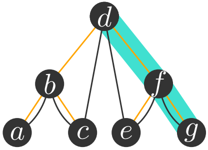
Treedepth vs. Treewidth

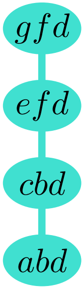
> The treewidth of a graph is as most as large as the treedepth.

This means graph classes with bounded treewidth are more general than those with bounded treedepth.

Proof: The set of all paths from leafs to the root yield a tree decomposition.





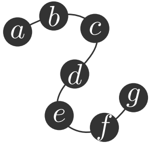


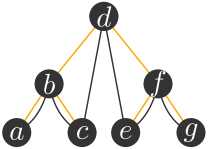
Treedepth vs. Treewidth

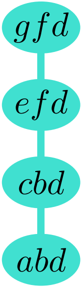
The treewidth of a graph is as most as large as the treedepth.

This means graph classes with bounded treewidth are more general than those with bounded treedepth.

Proof: The set of all paths from leafs to the root yield a tree decomposition.

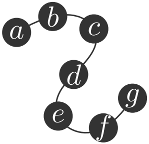


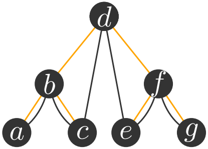




Treedepth of Paths

A treedepth of a path with n vertices is exactly $\lceil \log(n+1) \rceil$.





Low Treedepth Colorings

Nešetřil, Ossona de Mendez

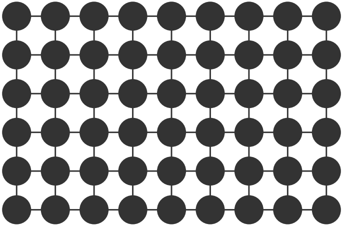
A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph?

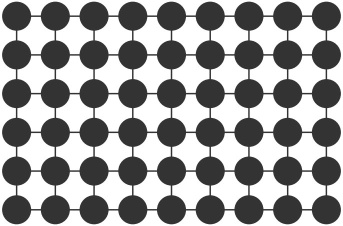


Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph? How many colors do we need here such that every set of 2 colors has treedepth ≤ 2?

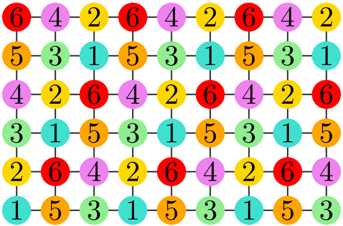


Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph? How many colors do we need here such that every set of 2 colors has treedepth ≤ 2?

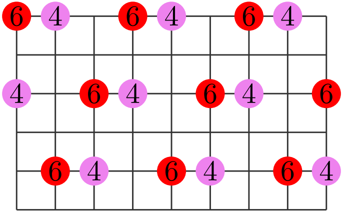


Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph? How many colors do we need here such that every set of 2 colors has treedepth ≤ 2?

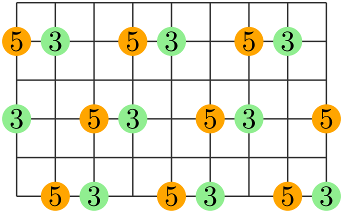


Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph? How many colors do we need here such that every set of 2 colors has treedepth ≤ 2?

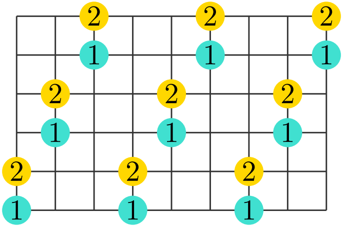


Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph? How many colors do we need here such that every set of 2 colors has treedepth ≤ 2?

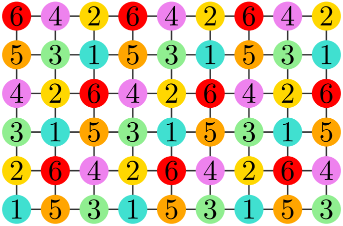


Low Treedepth Colorings

Nešetřil, Ossona de Mendez

A graph class $\mathcal{C}$ has bounded expansion iff there exists a function f such that for every $G \in \mathcal{C}$ and $p \in \mathbb{N}$ one can color G with $f(p)$ colors and every set of p colors induces a graph with treedepth $\leq p$.

What is $f(2)$ for this graph? How many colors do we need here such that every set of 2 colors has treedepth ≤ 2?



Low Treedepth Colorings

How many colors do we need to color a tree such that every set of p colors induces a graph with treedepth at most p?

Low Treedepth Colorings

How many colors do we need to color a tree such that every set of p colors induces a graph with treedepth at most p?

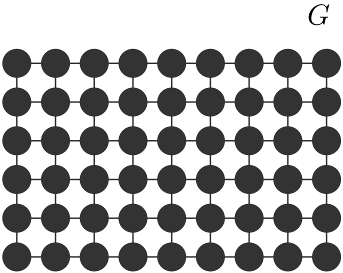
Color it with $p+1$ colors slicewise.

Existential Model-Checking

We can now use low-treedepth colorings to prove fpt existential model-checking.

Existential Model-Checking

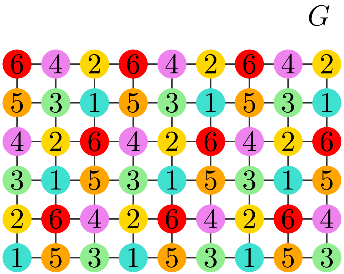
Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.



Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.

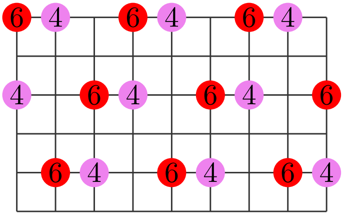


Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.

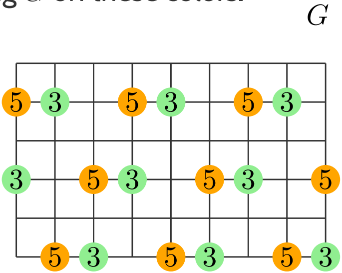
G



Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

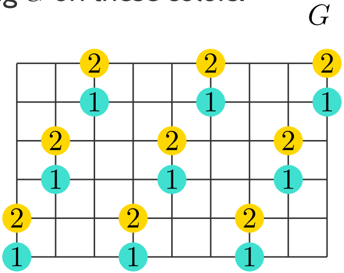
- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.



Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

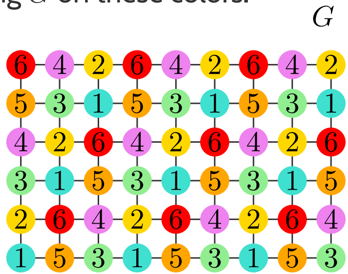
- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.



Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

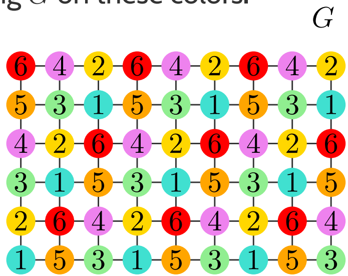
- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.



Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.
- We consider $\binom{f(|H|)}{|H|}$ color sets and for each we search for H in time $g(|H|) \cdot n$ using Courcelle.

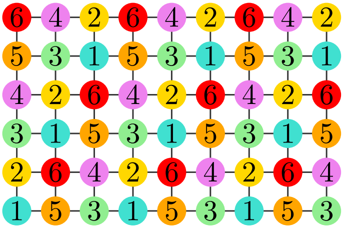


Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.
- We consider $\binom{f(|H|)}{|H|}$ color sets and for each we search for H in time $g(|H|) \cdot n$ using Courcelle.
- Total run time $\binom{f(|H|)}{|H|} g(|H|) \cdot n$.

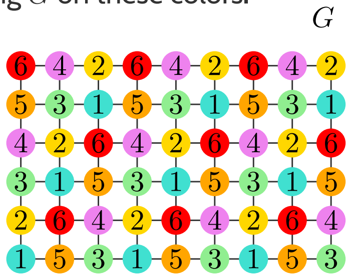
G



Existential Model-Checking

Let $\mathcal{C}$ be a class with bounded expansion, having low treedepth colorings with function $f(p)$. We want to know in time $h(|H|) \cdot n$ whether a graph $G \in \mathcal{C}$ has H as induced subgraph.

- For $p = |H|$ compute low treedepth coloring of G with $f(p)$ colors.
- H occurs in G iff there exists a set of $|H|$ colors such that H occurs in graph obtained by inducing G on these colors.
- We consider $\binom{f(|H|)}{|H|}$ color sets and for each we search for H in time $g(|H|) \cdot n$ using Courcelle.
- Total run time $\binom{f(|H|)}{|H|} g(|H|) \cdot n$.
- Plus time needed to compute coloring!



A Note on Nowhere Dense Classes

General rule: Things that work for bounded expansion also work for nowhere dense, but in an uglier way.

A Note on Nowhere Dense Classes

General rule: Things that work for bounded expansion also work for nowhere dense, but in an uglier way.

Nowhere dense classes can also be characterized via low treedepth colorings.

Nešetřil, Ossona de Mendez

If a graph class $\mathcal{C}$ is nowhere dense then there exists a function f such that for every $G \in \mathcal{C}$, every $\varepsilon > 0$, and $p \in \mathbb{N}$ one can color G with $f(\varepsilon, p)|n|^{\varepsilon}$ colors and every set of p colors induces a graph with treedepth $\leq p$.

Comparison

Bounded Expansion

Nowhere Dense

Comparison

Bounded Expansion

Nowhere Dense

p-treedepth colorings with $f(p)$ colors ✓

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^{\varepsilon}$ colors for all $\varepsilon > 0$ ✓

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Enumerate p-subsets in time $\binom{f(p)}{p}$ ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^{\varepsilon}$ colors for all $\varepsilon > 0$ ✓

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Enumerate p-subsets in time $\binom{f(p)}{p}$ ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^{\varepsilon}$ colors for all $\varepsilon > 0$ ✓

Enumerate p-subsets in time $\binom{f(\varepsilon,p)n^{\varepsilon}}{p}$

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Enumerate p-subsets in time $\binom{f(p)}{p}$ ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^\varepsilon$ colors for all $\varepsilon > 0$ ✓

Enumerate p-subsets in time $\binom{f(\varepsilon,p)n^\varepsilon}{p}$
$\leq f(\varepsilon, p)n^{\varepsilon p} = f(\varepsilon/p, p)n^\varepsilon$ ✓

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Enumerate p-subsets in time $\binom{f(p)}{p}$ ✓

Do something very expensive in time $2^{2^{f(p)}}$ ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^{\varepsilon}$ colors for all $\varepsilon > 0$ ✓

Enumerate p-subsets in time $\binom{f(\varepsilon,p)n^{\varepsilon}}{p}$
$\leq f(\varepsilon, p)n^{\varepsilon p} = f(\varepsilon/p, p)n^{\varepsilon}$ ✓

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Enumerate p-subsets in time $\binom{f(p)}{p}$ ✓

Do something very expensive in time $2^{2^{f(p)}}$ ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^{\varepsilon}$ colors for all $\varepsilon > 0$ ✓

Enumerate p-subsets in time $\binom{f(\varepsilon,p)n^{\varepsilon}}{p}$
$\leq f(\varepsilon, p)n^{\varepsilon p} = f(\varepsilon/p, p)n^{\varepsilon}$ ✓

Do something very expensive in time $2^{2^{f(\varepsilon/p,p)n^{\varepsilon}}}$

Comparison

Bounded Expansion

p-treedepth colorings with $f(p)$ colors ✓

Enumerate p-subsets in time $\binom{f(p)}{p}$ ✓

Do something very expensive in time $2^{2^{f(p)}}$ ✓

Nowhere Dense

p-treedepth colorings with $f(\varepsilon, p)n^{\varepsilon}$ colors for all $\varepsilon > 0$ ✓

Enumerate p-subsets in time $\binom{f(\varepsilon,p)n^{\varepsilon}}{p}$
$\leq f(\varepsilon, p)n^{\varepsilon p} = f(\varepsilon/p, p)n^{\varepsilon}$ ✓

Do something very expensive in time $2^{2^{f(\varepsilon/p,p)n^{\varepsilon}}}$
$\geq 2^{2^{\log(n)}} = 2^n$ ✗