Algorithmic Meta-Theorems

192.122
WS21/22

Jan Dreier
dreier@ac.tuwien.ac.at

ALGORITHMS AND COMPLEXITY GROUP
TU WIEN Informatics

Ehrenfeucht–Fraïssé Games

Certain properties cannot be decided in first-order logic. For example, there is no first-order formula φ such that $G \models \varphi$ iff G is connected. How do we prove that?

Ehrenfeucht–Fraïssé Games

Certain properties cannot be decided in first-order logic. For example, there is no first-order formula φ such that $G \models \varphi$ iff G is connected. How do we prove that?

- The *quantifier-rank* of a formula is the maximum number of nested quantifiers.

Ehrenfeucht–Fraïssé Games

Certain properties cannot be decided in first-order logic. For example, there is no first-order formula φ such that $G \models \varphi$ iff G is connected. How do we prove that?

- The *quantifier-rank* of a formula is the maximum number of nested quantifiers.
- We write $G \equiv_q G'$ if for all first-order sentences φ of quantifier-rank $\leq q$ holds $G \models \varphi \iff G' \models \varphi$.

Ehrenfeucht–Fraïssé Games

Certain properties cannot be decided in first-order logic. For example, there is no first-order formula φ such that $G \models \varphi$ iff G is connected. How do we prove that?

- The *quantifier-rank* of a formula is the maximum number of nested quantifiers.
- We write $G \equiv_q G'$ if for all first-order sentences φ of quantifier-rank $\leq q$ holds $G \models \varphi \iff G' \models \varphi$.
- Show that for every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.

Ehrenfeucht–Fraïssé Games

Certain properties cannot be decided in first-order logic. For example, there is no first-order formula φ such that $G \models \varphi$ iff G is connected. How do we prove that?

- The *quantifier-rank* of a formula is the maximum number of nested quantifiers.
- We write $G \equiv_q G'$ if for all first-order sentences φ of quantifier-rank $\leq q$ holds $G \models \varphi \iff G' \models \varphi$.
- Show that for every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.
- If there was a formula to decide connectivity it would have quantifier-rank q for some q. But this formula cannot tell G and G' apart. A contradiction.

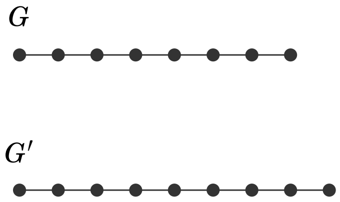
Ehrenfeucht–Fraïssé Games

Certain properties cannot be decided in first-order logic. For example, there is no first-order formula φ such that $G \models \varphi$ iff G is connected. How do we prove that?

- The *quantifier-rank* of a formula is the maximum number of nested quantifiers.
- We write $G \equiv_q G'$ if for all first-order sentences φ of quantifier-rank $\leq q$ holds $G \models \varphi \iff G' \models \varphi$.
- Show that for every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.
- If there was a formula to decide connectivity it would have quantifier-rank q for some q. But this formula cannot tell G and G' apart. A contradiction.
- Show $G \equiv_q G'$ using *Ehrenfeucht–Fraïssé games*.

Ehrenfeucht–Fraïssé Games

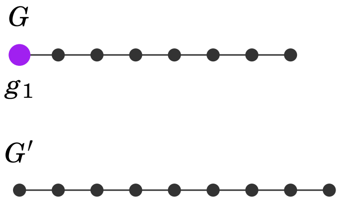
The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

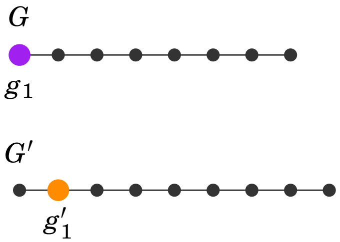
- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$



Ehrenfeucht–Fraïssé Games

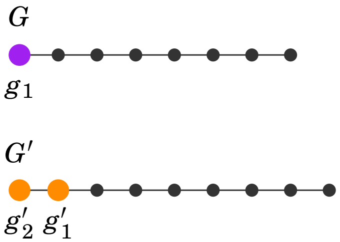
The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$
- Duplicator picks partner vertex in other graph.



The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

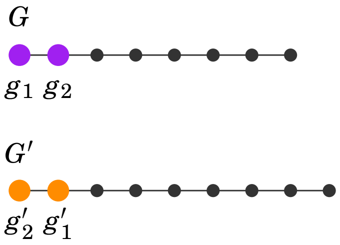
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

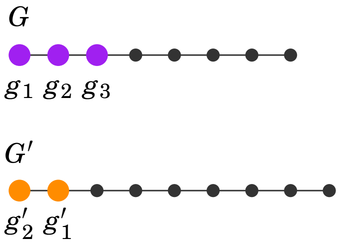
- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g'_1, \ldots, g'_q \in V(G')$ (pairwise distinct).



Ehrenfeucht–Fraïssé Games

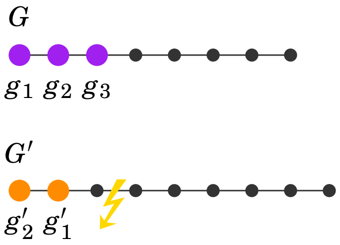
The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g'_1, \ldots, g'_q \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g'_i \sim g'_j$ for all i, j.



The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

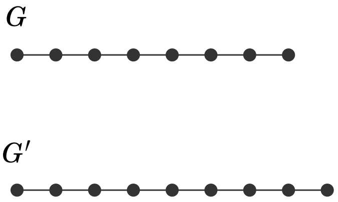
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

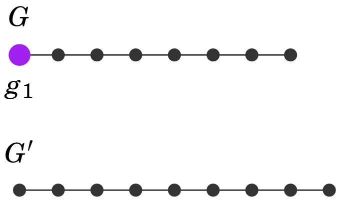
The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

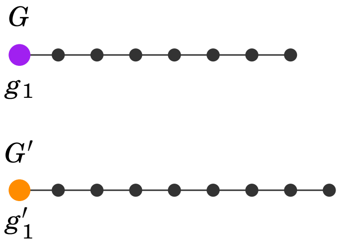
- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g'_1, \ldots, g'_q \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g'_i \sim g'_j$ for all i, j.



Ehrenfeucht–Fraïssé Games

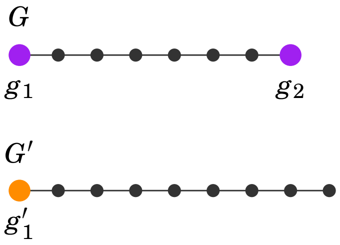
The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g'_1, \ldots, g'_q \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g'_i \sim g'_j$ for all i, j.



The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

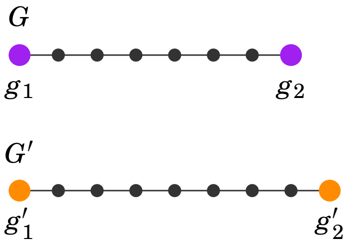
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

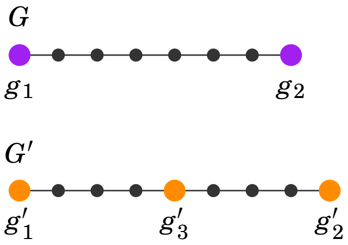
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

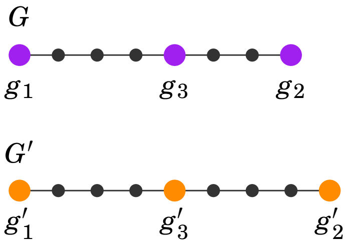
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

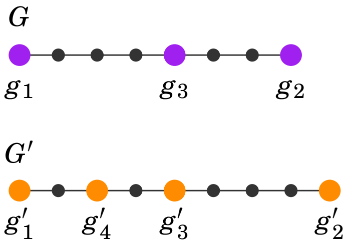
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

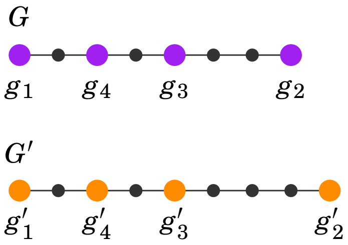
The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

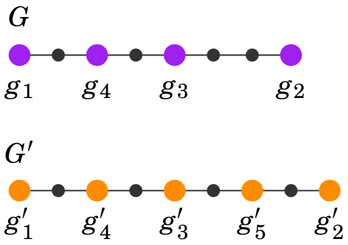
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

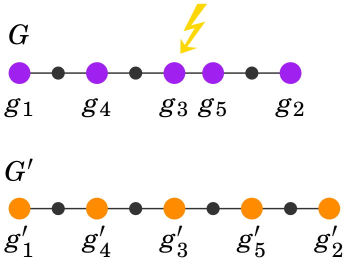
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

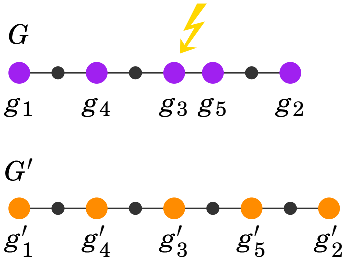
- Spoiler picks $g_i \in V(G)$ or $g_i' \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g_1', \ldots, g_q' \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g_i' \sim g_j'$ for all i, j.



Ehrenfeucht–Fraïssé Games

The q-round Ehrenfeucht–Fraïssé game between the *Duplicator* and the *Spoiler* is played on two graphs G and G'.

- Spoiler picks $g_i \in V(G)$ or $g'_i \in V(G')$
- Duplicator picks partner vertex in other graph.
- Repeat q times to get $g_1, \ldots, g_q \in V(G)$ and $g'_1, \ldots, g'_q \in V(G')$ (pairwise distinct).
- Duplicator wins if $g_i \sim g_j \iff g'_i \sim g'_j$ for all i, j.



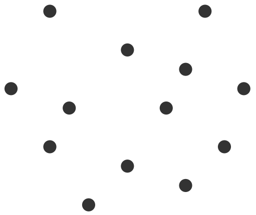
Theorem

$G \equiv_q G'$ iff the Duplicator wins the q-round Ehrenfeucht–Fraïssé game.

Parity

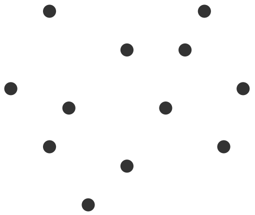
For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.

G



$q+1$ vertices

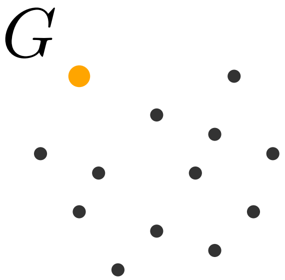
G'



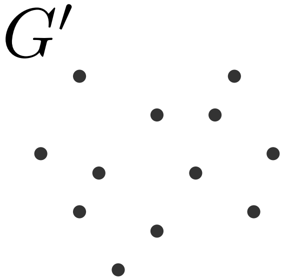
q vertices

Parity

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.



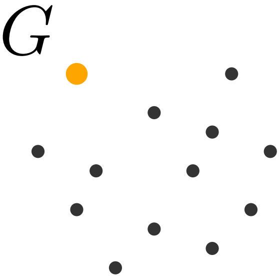
$q + 1$ vertices

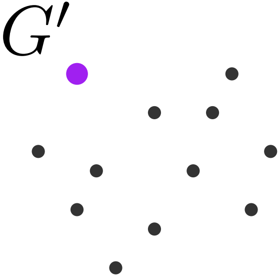


q vertices

Parity

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.



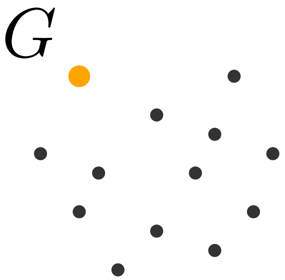


$q+1$ vertices

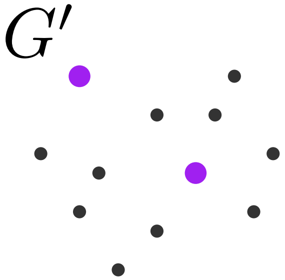
q vertices

Parity

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.



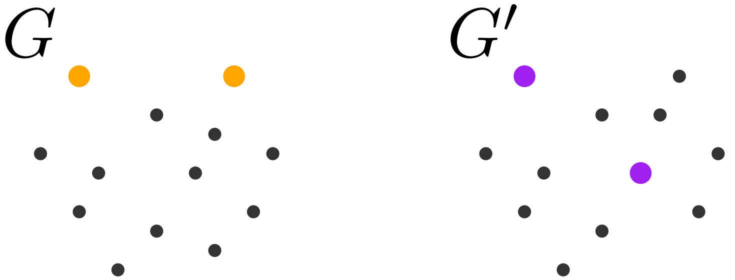
$q+1$ vertices



q vertices

Parity

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.



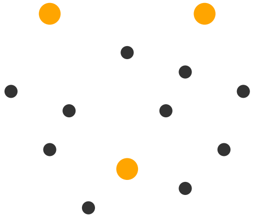
$q+1$ vertices

q vertices

Parity

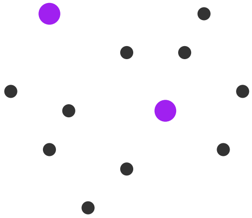
For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.

G



$q+1$ vertices

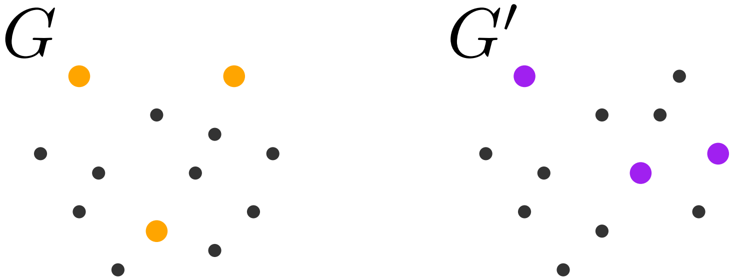
G'



q vertices

Parity

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.

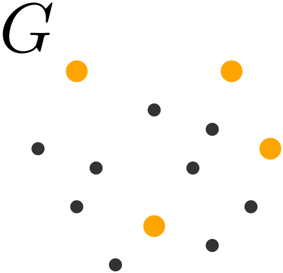


$q+1$ vertices

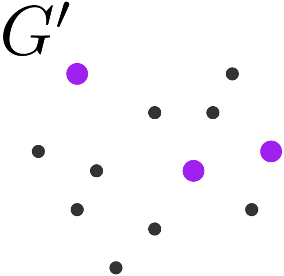
q vertices

Parity

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.

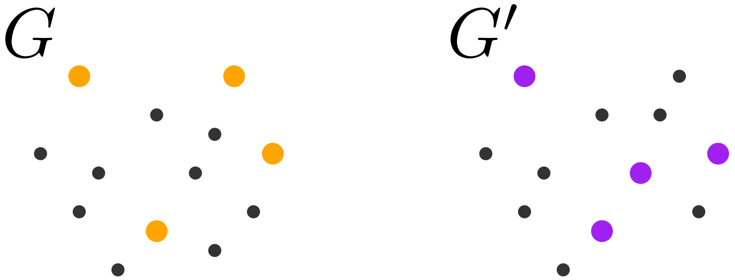


$q+1$ vertices



q vertices

For every q there is G with an even number of vertices and G' with an odd number of vertices and $G \equiv_q G'$.

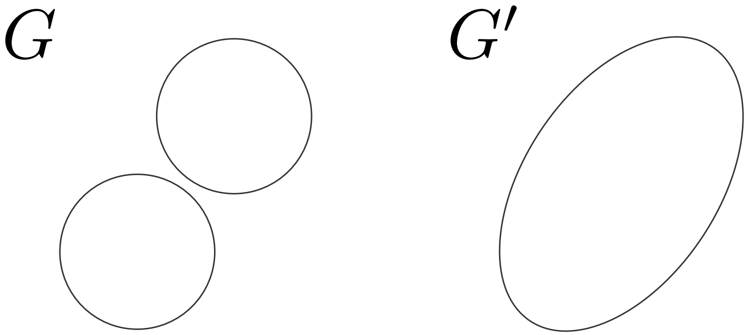


$q + 1$ vertices

q vertices

Connectivity

For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.

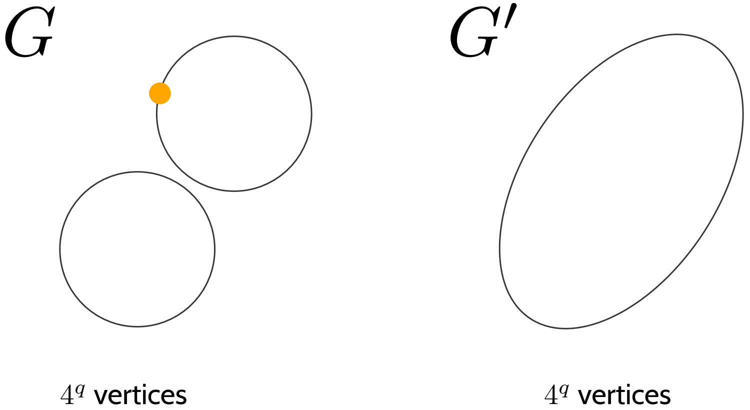


4^q vertices

4^q vertices

Connectivity

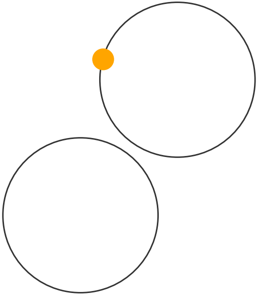
For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



Connectivity

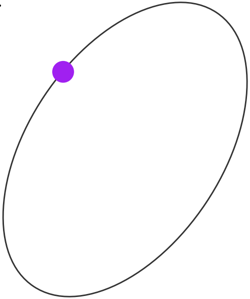
For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.

G



4^q vertices

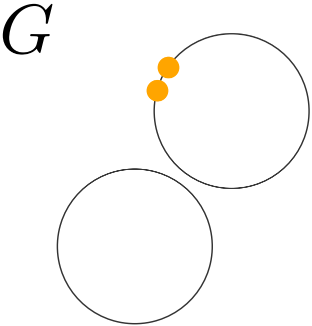
G'

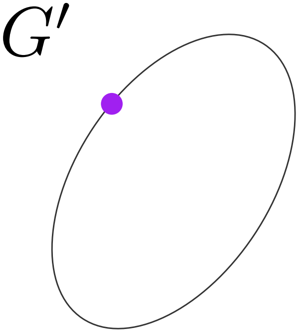


4^q vertices

Connectivity

For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



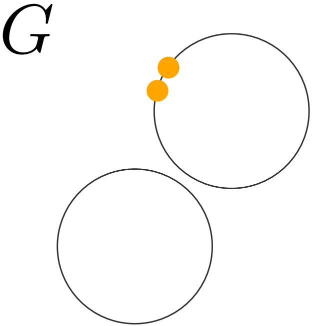


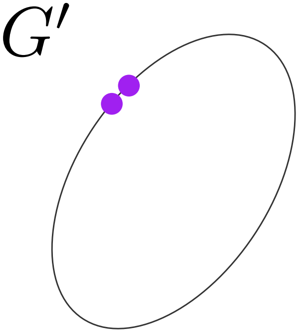
4^q vertices

4^q vertices

Connectivity

For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



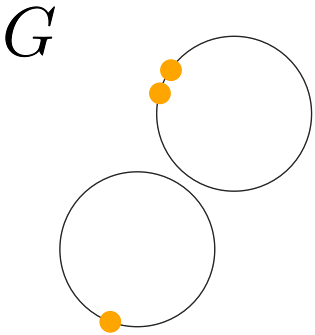


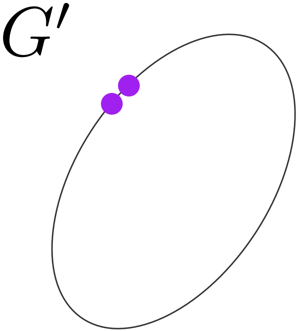
4^q vertices

4^q vertices

Connectivity

For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



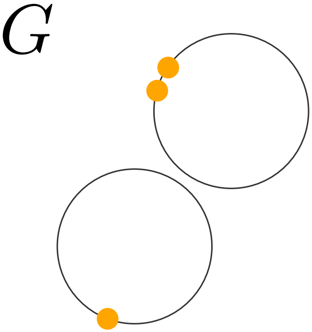


4^q vertices

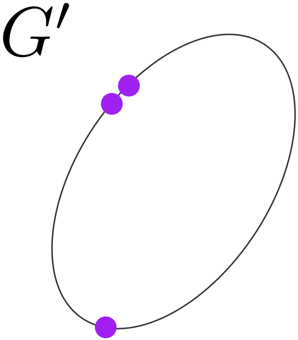
4^q vertices

Connectivity

For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



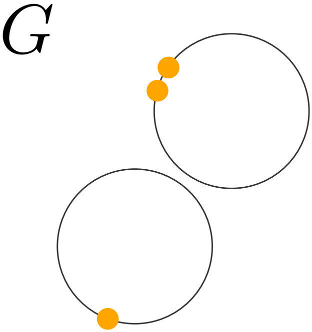
4^q vertices



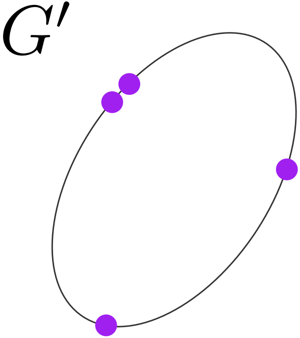
4^q vertices

Connectivity

For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



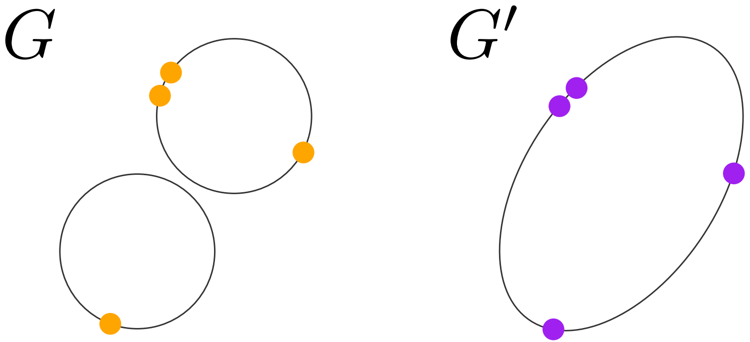
4^q vertices



4^q vertices

Connectivity

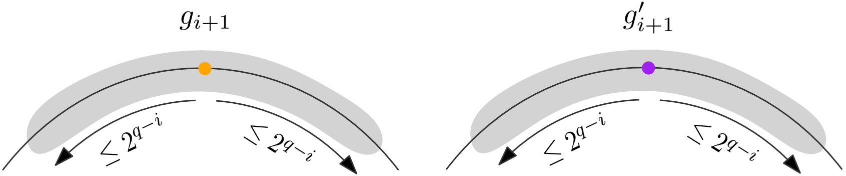
For every q there is a connected graph G and a disconnected graph G' with $G \equiv_q G'$.



4^q vertices 4^q vertices

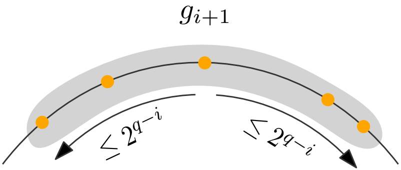
Connectivity

Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds $\text{dist}(g_j, g_k) =_{2^{q-i+1}} \text{dist}(g'_j, g'_k)$. Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i + 1$.



Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds dist$(g_j, g_k) =_{2^{q-i+1}}$ dist(g'_j, g'_k). Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i + 1$.

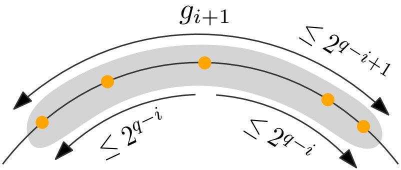
Case 1: g_{i+1} is close to other vertices



Connectivity

Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds dist$(g_j, g_k) =_{2^{q-i+1}}$ dist(g'_j, g'_k). Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i+1$.

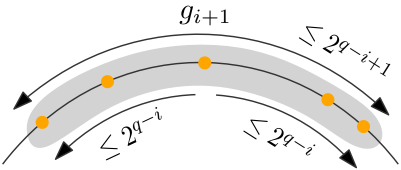
Case 1: g_{i+1} is close to other vertices

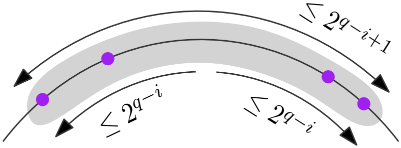


Connectivity

Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds dist$(g_j, g_k) =_{2^{q-i+1}}$ dist(g'_j, g'_k). Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i+1$.

Case 1: g_{i+1} is close to other vertices

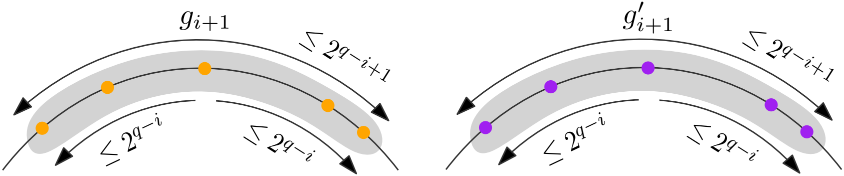




Connectivity

Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds dist$(g_j, g_k) =_{2^{q-i+1}}$ dist(g'_j, g'_k). Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i+1$.

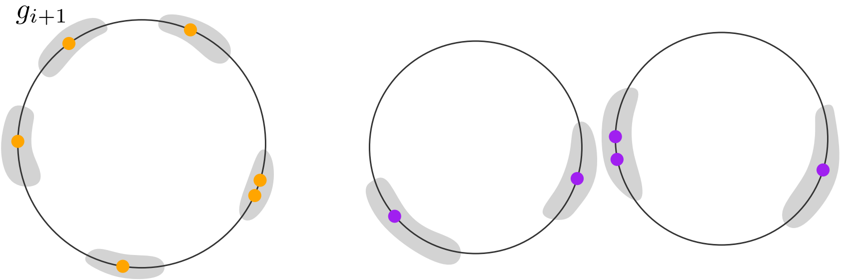
Case 1: g_{i+1} is close to other vertices



Connectivity

Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds dist$(g_j, g_k) =_{2^{q-i+1}}$ dist(g'_j, g'_k). Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i + 1$.

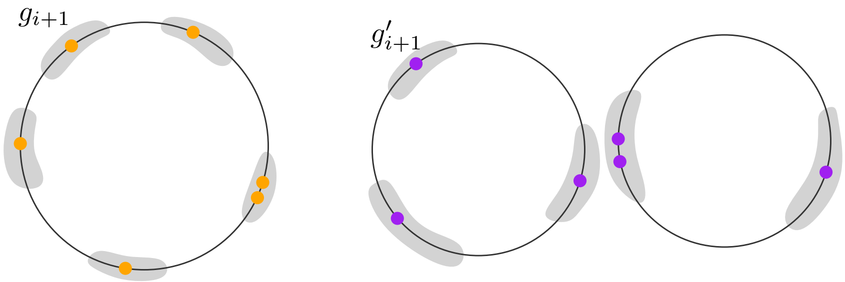
Case 2: g_{i+1} is not close to other vertices



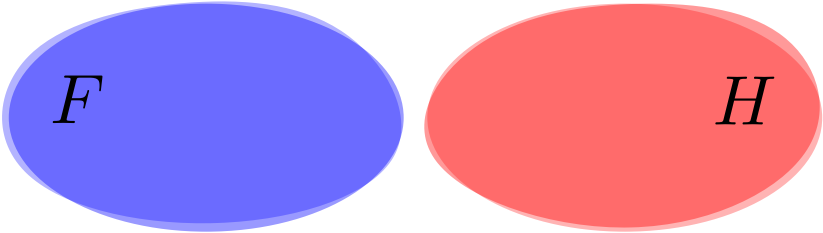
Connectivity

Proof by induction: The Duplicator can play such that for elements $g_1, \ldots, g_i$ and $g'_1, \ldots, g'_i$ choosen after i rounds dist$(g_j, g_k) =_{2^{q-i+1}}$ dist(g'_j, g'_k). Here $a =_c b$ means that a and b are equal or both larger than c. Show that this also holds for $i+1$.

Case 2: g_{i+1} is not close to other vertices

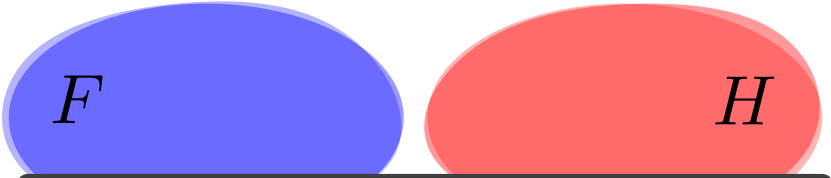


q-type(H) is the set of all FO-sentences ξ of quantifier-rank $\leq q$ with $H \models \xi$. Note that q-type$(F) = q$-type(H) iff $F \equiv_q H$.



Fefermann–Vaught (First-Order)

q-type(H) is the set of all FO-sentences ξ of quantifier-rank $\leq q$ with $H \models \xi$. Note that q-type$(F) = q$-type(H) iff $F \equiv_q H$.

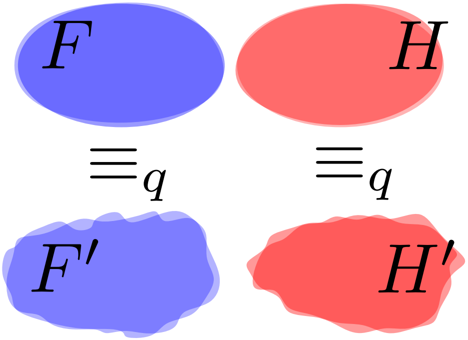


Theorem (Fefermann–Vaught)

Let F and H be graphs with $V(F) \cap V(H) = \emptyset$.
Then q-type$(F \cup H)$ is completely determined by

- q-type(F) and
- q-type(H).

Fefermann–Vaught (First-Order)



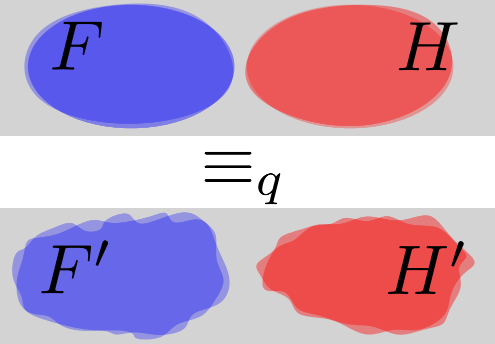
Alternative Formulation

Let F, F', H, H' be graphs. If

- $F \equiv_q F'$ and
- $H \equiv_q H'$

then $F \cup H \equiv_q F' \cup H'$.

Fefermann–Vaught (First-Order)



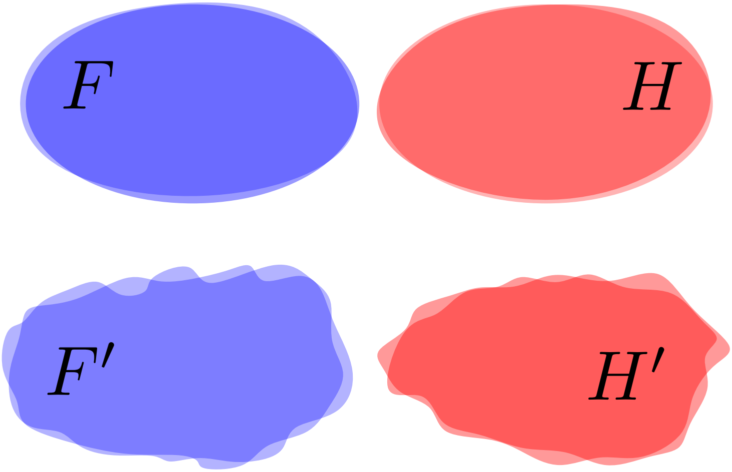
Alternative Formulation

Let F, F', H, H' be graphs. If

- $F \equiv_q F'$ and
- $H \equiv_q H'$

then $F \cup H \equiv_q F' \cup H'$.

Proof: Combine the two winning strategies of the Duplicator.



Summary

- First-order logic captures certain types of problems very well but fails spectacularly at some other very basic tasks.

Summary

- First-order logic captures certain types of problems very well but fails spectacularly at some other very basic tasks.
- There are more expressive extensions
  - FO + parity
  - FO + counting
  - FO + fixed points
  - …

  But model-checking may be harder for them.

Summary

- First-order logic captures certain types of problems very well but fails spectacularly at some other very basic tasks.
- There are more expressive extensions
  - FO + parity
  - FO + counting
  - FO + fixed points
  - …

  But model-checking may be harder for them.
- The field of *descriptive complexity* dedicated to studying the expressiveness of logics.

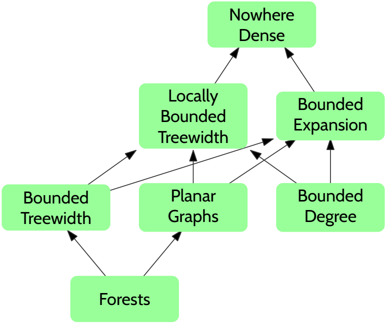
Our First First-Order Meta-Theorem

We will show:

Theorem (Seese 1996)

Let $\mathcal{C}$ be a graph class with bounded degree. 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 $f(|\varphi|)n$.

Later we will generalize this to other sparse graph classes.



Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k?

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k?

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.
- Finding a $k \times k$-grid?

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.
- Finding a $k \times k$-grid? Yes.

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.
- Finding a $k \times k$-grid? Yes.
- Deciding if a graph is connected?

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.
- Finding a $k \times k$-grid? Yes.
- Deciding if a graph is connected? No.

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.
- Finding a $k \times k$-grid? Yes.
- Deciding if a graph is connected? No.
- Finding a set of size k that dominates $n/2$ vertices?

Implications

If $\mathcal{C}$ is a class with bounded degree, which problems does the theorem solve in time $f(k) \cdot n$ on $\mathcal{C}$?

- Finding an independent set of size k? Yes.
- Finding an dominating set of size k? Yes.
- Finding a $k \times k$-grid? Yes.
- Deciding if a graph is connected? No.
- Finding a set of size k that dominates $n/2$ vertices? No.

Proof Idea

We show how to …

- find things that are far apart (independent sets, …),
- find things that are local (connected subgraphs, …).

Proof Idea

We show how to …

- find things that are far apart (independent sets, …),
- find things that are local (connected subgraphs, …).

A famous logic theorem by Gaifman shows that every problem definable in first-order logic can be decomposed into subproblems from these two categories.

Proof Idea

We show how to …

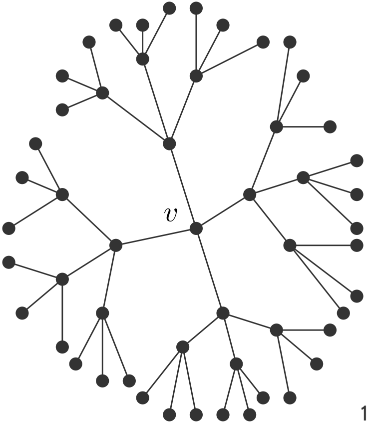
- find things that are far apart (independent sets, …),
- find things that are local (connected subgraphs, …).

A famous logic theorem by Gaifman shows that every problem definable in first-order logic can be decomposed into subproblems from these two categories.

One can use the same technique for *locally bounded treewidth* graph classes. This includes: planar graphs, bounded treewidth, bounded degree, bounded genus, …

Main Tool: Bounded Neighborhood Size

Let v be a vertex in a graph with maximal degree d. Then $|N_r(v)| \leq d^{r+1}$.

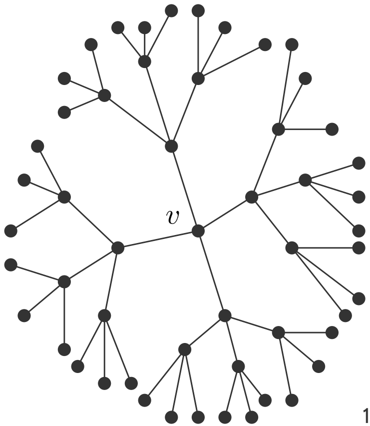


Main Tool: Bounded Neighborhood Size

Let v be a vertex in a graph with maximal degree d. Then $|N_r(v)| \leq d^{r+1}$.

Proof:

- We show there are at most d^i vertices with distance i. Then $\sum_{i=1}^{r} d^i \leq d^{r+1}$.

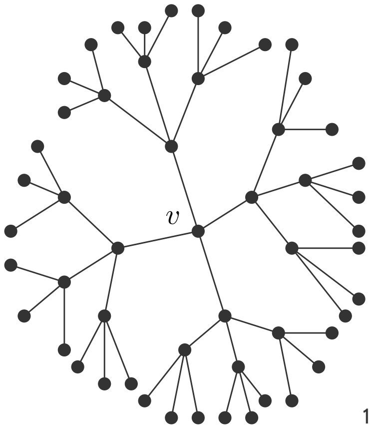


Main Tool: Bounded Neighborhood Size

Let v be a vertex in a graph with maximal degree d. Then $|N_r(v)| \leq d^{r+1}$.

Proof:

- We show there are at most d^i vertices with distance i. Then $\sum_{i=1}^{r} d^i \leq d^{r+1}$.
- There is $1 = (d-1)^0$ vertex with distance 0.

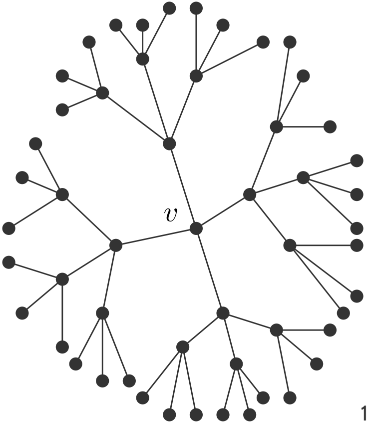


Main Tool: Bounded Neighborhood Size

Let v be a vertex in a graph with maximal degree d. Then $|N_r(v)| \leq d^{r+1}$.

Proof:

- We show there are at most d^i vertices with distance i. Then $\sum_{i=1}^{r} d^i \leq d^{r+1}$.
- There is $1 = (d-1)^0$ vertex with distance 0.
- Number can increase at most by factor of d as we go from i to $i+1$.



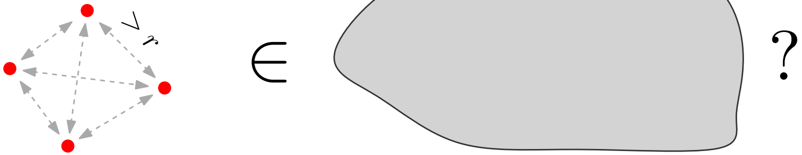
Finding Things That Are Far Appart

Let $\mathcal{C}$ be a graph class with bounded degree. One can decide in time $f(r,k)n$ whether a graph $G \in \mathcal{C}$ contains a red r-scattered set of size k.

Finding Things That Are Far Appart

red r-scattered set:
k red vertices with
pairwise distance $> r$

degree $\leq d$

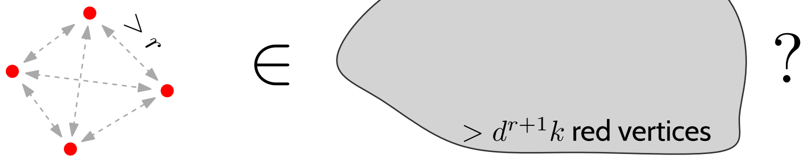


Is there a red r-scattered set of size k in a graph of degree $\leq d$?

Finding Things That Are Far Appart

red r-scattered set:
k red vertices with
pairwise distance $> r$

degree $\leq d$

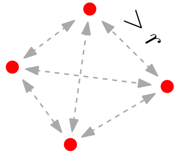


Is there a red r-scattered set of size k in a graph of degree $\leq d$?

- Assume we have more than $d^{r+1}k$ red vertices. Otherwise solve using brute force in time $(d^{r+1}k)^{O(k)}$.

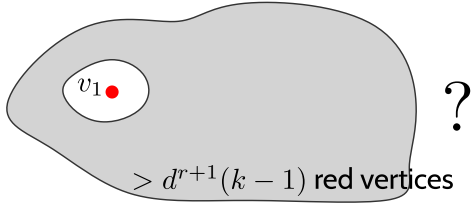
Finding Things That Are Far Appart

red r-scattered set:
k red vertices with
pairwise distance $> r$



$\in$

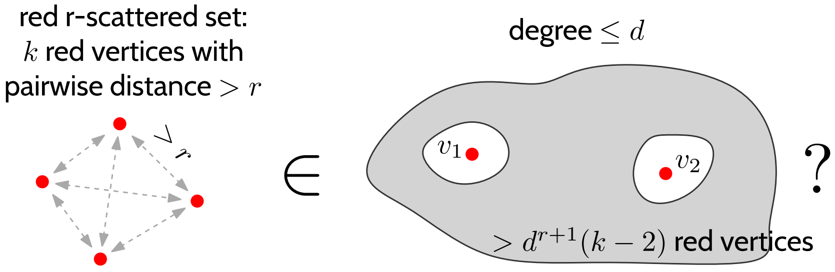
degree $\leq d$



Is there a red r-scattered set of size k in a graph of degree $\leq d$?

- Assume we have more than $d^{r+1}k$ red vertices. Otherwise solve using brute force in time $(d^{r+1}k)^{O(k)}$.
- Greedily pick a vertex v_1 and remove $N_r(v_1)$ (of size $\leq d^{r+1}$)

Finding Things That Are Far Appart



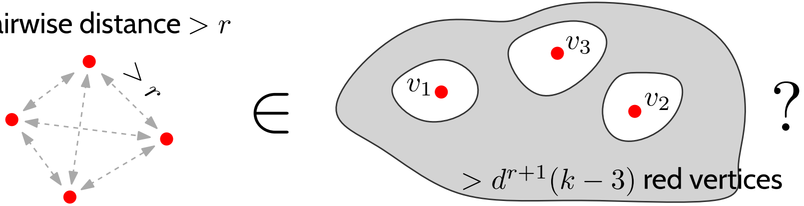
Is there a red r-scattered set of size k in a graph of degree $\leq d$?

- Assume we have more than $d^{r+1}k$ red vertices. Otherwise solve using brute force in time $(d^{r+1}k)^{O(k)}$.
- Greedily pick a vertex v_1 and remove $N_r(v_1)$ (of size $\leq d^{r+1}$)
- Repeat k times. This is a solution.

Finding Things That Are Far Appart

red r-scattered set:
k red vertices with
pairwise distance $> r$

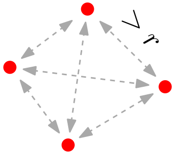
degree $\leq d$



Is there a red r-scattered set of size k in a graph of degree $\leq d$?

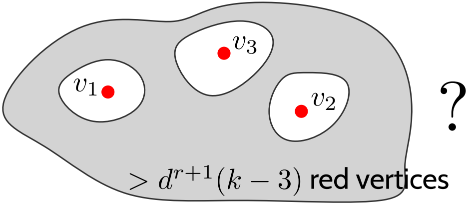
- Assume we have more than $d^{r+1}k$ red vertices. Otherwise solve using brute force in time $(d^{r+1}k)^{O(k)}$.
- Greedily pick a vertex v_1 and remove $N_r(v_1)$ (of size $\leq d^{r+1}$)
- Repeat k times. This is a solution.
- Works because $\leq d^{r+1}$ vertices are removed each round.

red r-scattered set:
k red vertices with
pairwise distance $> r$



$\in$

degree $\leq d$



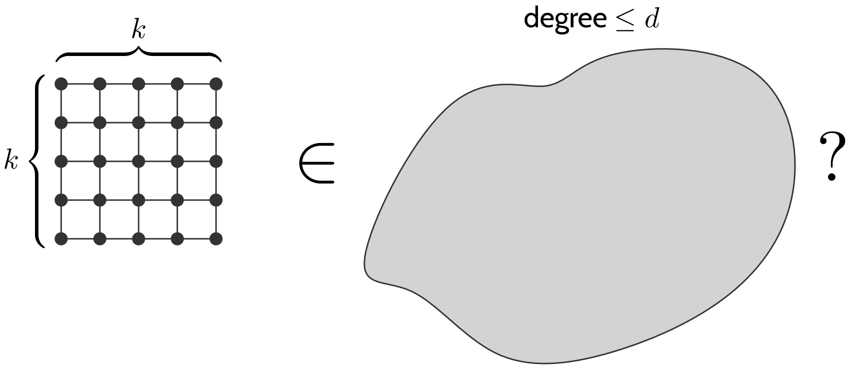
Is there a red r-scattered set of size k in a graph of degree $\leq d$?

- Assume we have more than $d^{r+1}k$ red vertices. Otherwise solve using brute force in time $(d^{r+1}k)^{O(k)}$.
- Greedily pick a vertex v_1 and remove $N_r(v_1)$ (of size $\leq d^{r+1}$)
- Repeat k times. This is a solution.
- Works because $\leq d^{r+1}$ vertices are removed each round.
- Run time: $(d^{r+1}k)^{O(k)}n$.

Finding Things That Are Local

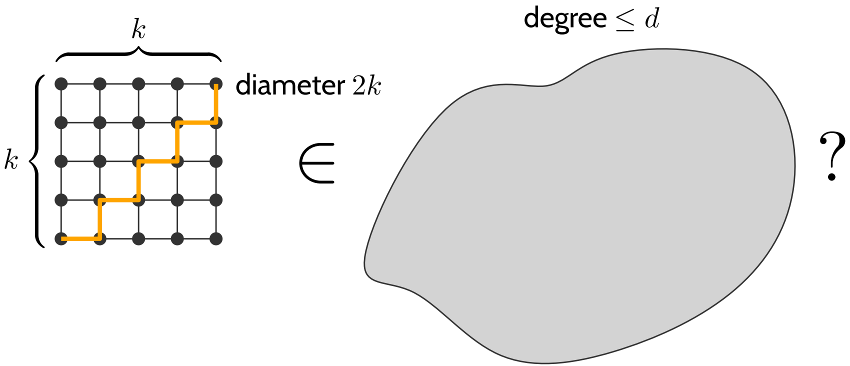
Let $\mathcal{C}$ be a graph class with bounded degree. One can decide in time $f(k)n$ whether a graph $G \in \mathcal{C}$ contains a $k \times k$-grid as an induced subgraph.

Finding Things That Are Local



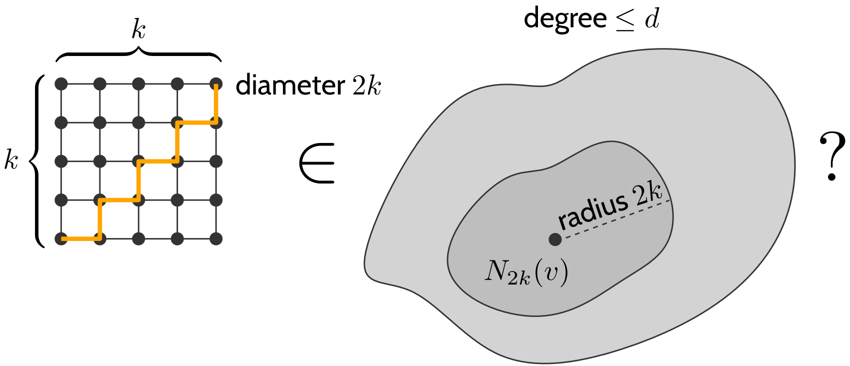
- Is there a $k \times k$ grid in a graph of degree $\leq d$?

Finding Things That Are Local



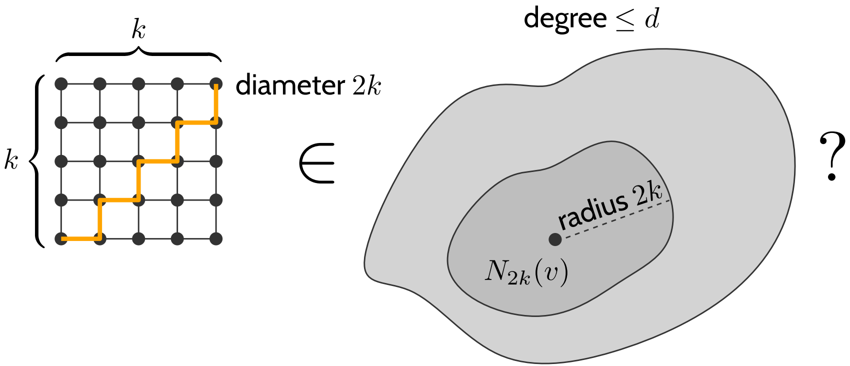
- Is there a $k \times k$ grid in a graph of degree $\leq d$?

Finding Things That Are Local



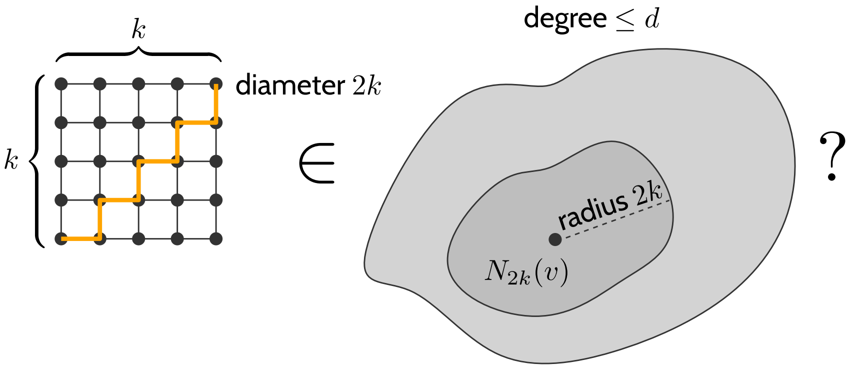
- Is there a $k \times k$ grid in a graph of degree $\leq d$?
- If yes, then it is contained in $N_{2r}(v)$ for some vertex v.

Finding Things That Are Local



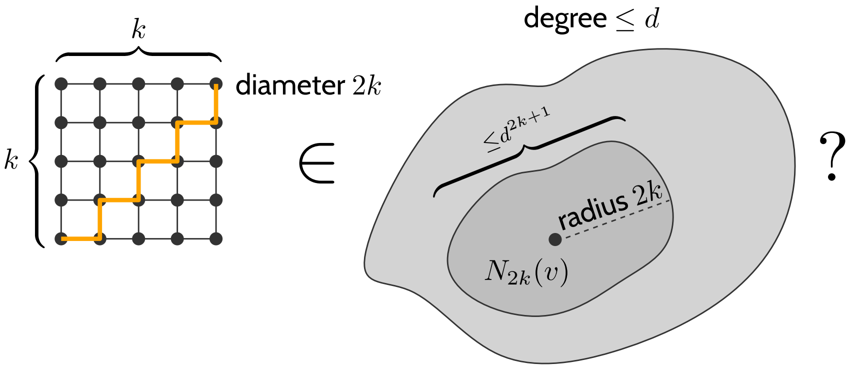
- Is there a $k \times k$ grid in a graph of degree $\leq d$?
- If yes, then it is contained in $N_{2r}(v)$ for some vertex v.
- Enumerate every vertex v.

Finding Things That Are Local



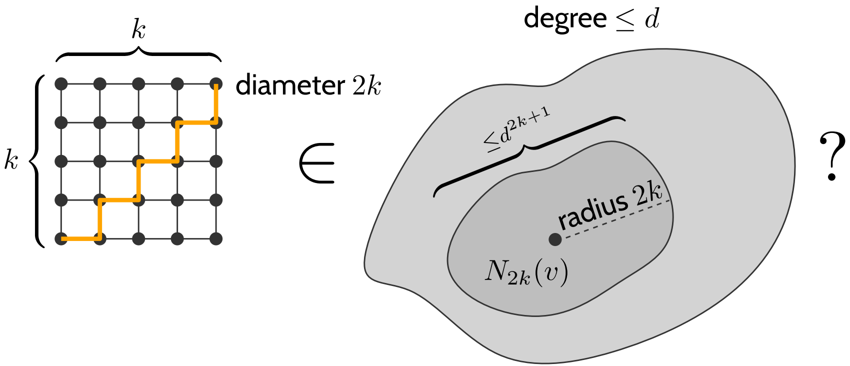
- Is there a $k \times k$ grid in a graph of degree $\leq d$?
- If yes, then it is contained in $N_{2r}(v)$ for some vertex v.
- Enumerate every vertex v.
- Check in time $|N_{2r}(v)|^{O(k^2)}$ whether neighborhood has grid.

Finding Things That Are Local



- Is there a $k \times k$ grid in a graph of degree $\leq d$?
- If yes, then it is contained in $N_{2r}(v)$ for some vertex v.
- Enumerate every vertex v.
- Check in time $|N_{2r}(v)|^{O(k^2)}$ whether neighborhood has grid.

Finding Things That Are Local



- Is there a $k \times k$ grid in a graph of degree $\leq d$?
- If yes, then it is contained in $N_{2r}(v)$ for some vertex v.
- Enumerate every vertex v.
- Check in time $|N_{2r}(v)|^{O(k^2)}$ whether neighborhood has grid.
- Total time: $d^{(2k+1)O(k^2)}n$

Combining Both Ideas

Homework

Let $\mathcal{C}$ be a graph class with bounded degree. For every graph H and $G \in \mathcal{C}$ one can decide in time $f(|H|)n$ whether G contains H as induced subgraph.

Combining Both Ideas

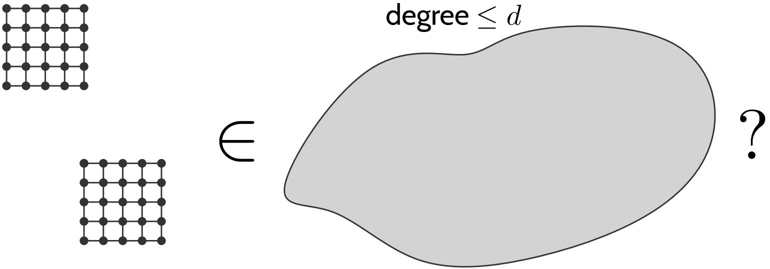
Homework

Let $\mathcal{C}$ be a graph class with bounded degree. For every graph H and $G \in \mathcal{C}$ one can decide in time $f(|H|)n$ whether G contains H as induced subgraph.

Now

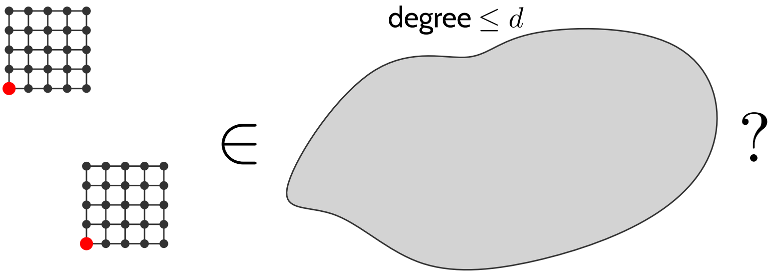
Let $\mathcal{C}$ be a graph class with bounded degree. One can decide in time $f(k)n$ whether a graph $G \in \mathcal{C}$ contains two disjoint $k \times k$-grids as induced subgraph.

Combining Both Ideas



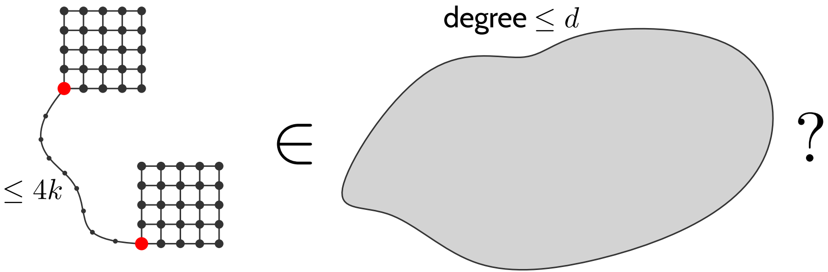
- Are there two disjoint $k \times k$ grids in a graph of degree $\leq d$?

Combining Both Ideas



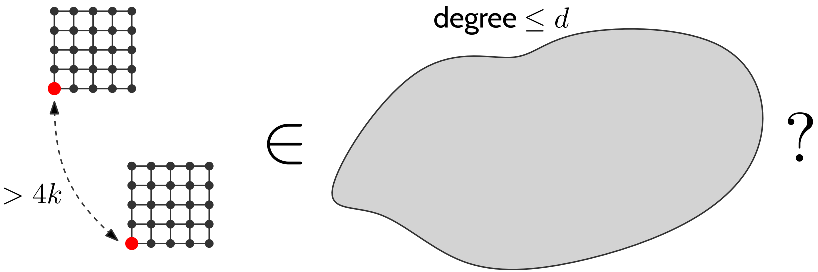
- Are there two disjoint $k \times k$ grids in a graph of degree $\leq d$?
- Mark one corner of each grid red.

Combining Both Ideas



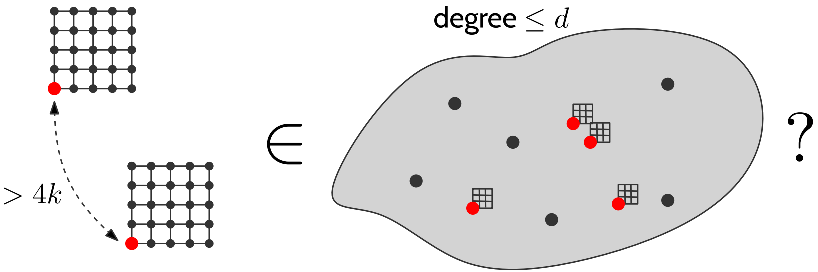
- Are there two disjoint $k \times k$ grids in a graph of degree $\leq d$?
- Mark one corner of each grid red.
- Case 1: There is a path of length $\leq 4k$ between red vertices.
  - Use existing aproach to find connected subgraph.

Combining Both Ideas



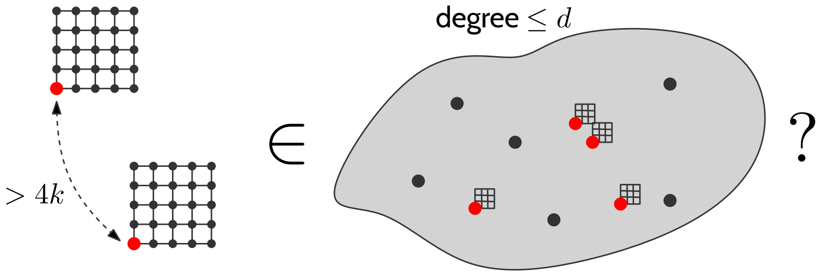
- Are there two disjoint $k \times k$ grids in a graph of degree $\leq d$?
- Mark one corner of each grid red.
- Case 1: There is a path of length $\leq 4k$ between red vertices.
  - Use existing aproach to find connected subgraph.
- Case 2: Red vertices have distance $> 4k$.

Combining Both Ideas



- Are there two disjoint $k \times k$ grids in a graph of degree $\leq d$?
- Mark one corner of each grid red.
- Case 1: There is a path of length $\leq 4k$ between red vertices.
  - Use existing aproach to find connected subgraph.
- Case 2: Red vertices have distance $> 4k$.
  - Mark all grid corners in the graph red (using brute force on neighborhoods).

Combining Both Ideas



- Are there two disjoint $k \times k$ grids in a graph of degree $\leq d$?
- Mark one corner of each grid red.
- Case 1: There is a path of length $\leq 4k$ between red vertices.
  - Use existing aproach to find connected subgraph.
- Case 2: Red vertices have distance $> 4k$.
  - Mark all grid corners in the graph red (using brute force on neighborhoods).
  - Use existing approach to find two red vertices with distance $> 4k$. Since grids have radius $2k$, their grids are disjoint.

Gaifman

The same kind of split into “local” and “far apart” parts can be done for first-order logic as well.

In a sense, all that first-order logic can do is say certain “local” properties are “far apart”.

This is formalized in Gaifman’s locality Theorem.

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\, x{\sim}y$

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\, x{\sim}y$ Yes

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\ x{\sim}y$ Yes
- $\omega(x) = \exists y\ x{\not\sim}y$

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\ x{\sim}y$ Yes
- $\omega(x) = \exists y\ x{\not\sim}y$ No

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\ x{\sim}y$ Yes
- $\omega(x) = \exists y\ x{\not\sim}y$ No
- $\omega(x)$ saying that x is corner of some $k \times k$-grid.

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\; x{\sim}y$ Yes
- $\omega(x) = \exists y\; x{\not\sim}y$ No
- $\omega(x)$ saying that x is corner of some $k \times k$-grid. Yes

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- ○ $\omega(x) = \exists y\ x{\sim}y$ Yes
- ○ $\omega(x) = \exists y\ x{\not\sim}y$ No
- ○ $\omega(x)$ saying that x is corner of some $k \times k$-grid. Yes
- ○ ... and there also is a second disjoint $k \times k$-grid.

Local Formulas

A formula $\omega(x)$ is called r-*local* if for all graphs G and all $v \in V(G)$

$$G \models \omega(v) \iff G[N_r(v)] \models \omega(v).$$

Are these formulas local for some r?

- $\omega(x) = \exists y\ x{\sim}y$ Yes
- $\omega(x) = \exists y\ x{\not\sim}y$ No
- $\omega(x)$ saying that x is corner of some $k \times k$-grid. Yes
- ... and there also is a second disjoint $k \times k$-grid. No

Evaluating Local Formulas

Let G be a graph with degree $\leq\ d$ and $\omega(x)$ be an r-local formula. One can compute in in time $O(d^{(r+1)|\varphi|}n)$ the set of all $v \in V(G)$ with $G \models \omega(v)$.

Evaluating Local Formulas

Let G be a graph with degree $\leq\ d$ and $\omega(x)$ be an r-local formula. One can compute in in time $O(d^{(r+1)|\varphi|}n)$ the set of all $v \in V(G)$ with $G \models \omega(v)$.

Proof:

- Enumerate all vertices v.
- Instead of evaluating $\varphi(v)$ on G, we evaluate it on $G[N_r(v)]$.
- $G[N_r(v)]$ contains $\leq d^{r+1}$ vertices.
- We can evaluate $\varphi(v)$ on a graph with k vertices in time $O(k^{|\varphi|})$.

Basic Local Sentences

Let $\omega(x)$ be an r-local formula.

Basic Local Sentences

Let $\omega(x)$ be an r-local formula. Let $\mathrm{dist}_{>2r}(x_1, x_2)$ be the first-order formula denoting that the distance between x_1 and x_2 is greater than $2r$.

Basic Local Sentences

Let $\omega(x)$ be an r-local formula. Let $\text{dist}_{>2r}(x_1, x_2)$ be the first-order formula denoting that the distance between x_1 and x_2 is greater than $2r$. A *basic local sentence* is a sentence of the form

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \text{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_{i} \omega(x_i) \Big).$$

"There are s vertices that all satisfy some local property ω and are far apart."

Example: There exist s vertices with degree 4 and pairwise distance larger than 10.

Gaifman's Locality Theorem

Every first-order sentence is equivalent to a boolean combination of basic local sentences.
For a given sentence, such a boolean combination can be computed.

Gaifman's Locality Theorem

Every first-order sentence is equivalent to a boolean combination of basic local sentences.
For a given sentence, such a boolean combination can be computed.

What is a boolean combination of basic local sentences expressing that there are two disjoint $k \times k$-grids?

Gaifman

Gaifman's Locality Theorem

Every first-order sentence is equivalent to a boolean combination of basic local sentences.
For a given sentence, such a boolean combination can be computed.

What is a boolean combination of basic local sentences expressing that there are two disjoint $k \times k$-grids?

- either there exists s_1 that is part of a subgraph □–□ ($6k$-local),

Gaifman

Gaifman's Locality Theorem

Every first-order sentence is equivalent to a boolean combination of basic local sentences.
For a given sentence, such a boolean combination can be computed.

What is a boolean combination of basic local sentences expressing that there are two disjoint $k \times k$-grids?

- either there exists s_1 that is part of a subgraph □–□ ($6k$-local),
- or there exist s_1, s_2, both part of a subgraph □ ($2k$-local) with $\text{dist}_{>4k}(s_1, s_2)$.

Proof of First-Order Meta-Theorem

Theorem (Seese 1996)

Let $\mathcal{C}$ be a graph class with bounded degree. 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 $f(|\varphi|)n$.

Proof of First-Order Meta-Theorem

Theorem (Seese 1996)

Let $\mathcal{C}$ be a graph class with bounded degree. 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 $f(|\varphi|)n$.

Proof:

- By Gaifman's theorem, φ is equivalent of boolean combination of basic local sentences such as

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathrm{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

  The size and number of these basic local sentences depends only on φ. We show how to evaluate such sentences.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_{i} \omega(x_i) \big).$$

Proof

Consider a basic local sentence

$$\exists x_1 \dots \exists x_s \big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \big).$$

We compute the set of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \big).$$

We compute the set of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have size $\leq d^{r+1}$.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have size $\leq d^{r+1}$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $O(d^{(r+1)|\varphi|})$ (brute force).

Proof

Consider a basic local sentence

$$\exists x_1 \dots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have size $\leq d^{r+1}$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $O(d^{(r+1)|\varphi|})$ (brute force).
- Doing it for all $v \in V(G)$ takes time $O(d^{(r+1)|\varphi|}n)$.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathrm{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have size $\leq d^{r+1}$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $O(d^{(r+1)|\varphi|})$ (brute force).
- Doing it for all $v \in V(G)$ takes time $O(d^{(r+1)|\varphi|} n)$.

We have to find a red $2r$-scattered set of size s in G. This can be done in time $f(r, s)n$ in bounded degree classes.

Locally Bounded Treewidth

Our approach for bounded degree easily generalizes to *locally bounded treewidth*.

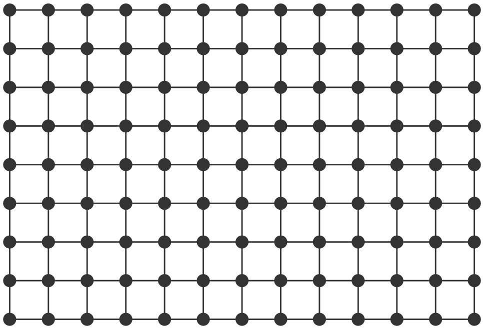
Locally Bounded Treewidth

A graph class $\mathcal{C}$ has *locally bounded treewidth* if there exists a function $g(r)$ such that for every graph $G \in \mathcal{C}$, every $v \in V(G)$ and every $r \in \mathbb{N}$ the neighborhood $G[N_r(v)]$ has treewidth at most $g(r)$.

Locally Bounded Treewidth

A graph class $\mathcal{C}$ has *locally bounded treewidth* if there exists a function $g(r)$ such that for every graph $G \in \mathcal{C}$, every $v \in V(G)$ and every $r \in \mathbb{N}$ the neighborhood $G[N_r(v)]$ has treewidth at most $g(r)$.

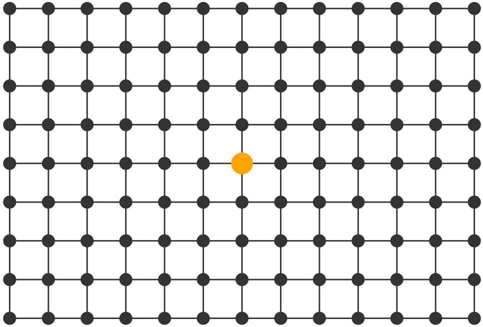
Which of these classes have locally bounded treewidth?



Locally Bounded Treewidth

A graph class $\mathcal{C}$ has *locally bounded treewidth* if there exists a function $g(r)$ such that for every graph $G \in \mathcal{C}$, every $v \in V(G)$ and every $r \in \mathbb{N}$ the neighborhood $G[N_r(v)]$ has treewidth at most $g(r)$.

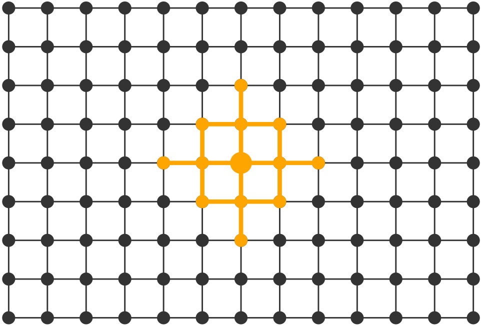
Which of these classes have locally bounded treewidth?



Locally Bounded Treewidth

A graph class $\mathcal{C}$ has *locally bounded treewidth* if there exists a function $g(r)$ such that for every graph $G \in \mathcal{C}$, every $v \in V(G)$ and every $r \in \mathbb{N}$ the neighborhood $G[N_r(v)]$ has treewidth at most $g(r)$.

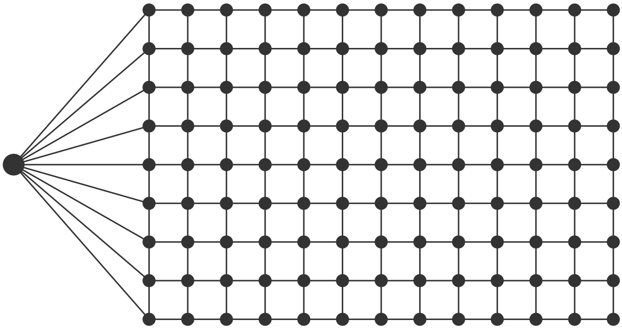
Which of these classes have locally bounded treewidth?



Locally Bounded Treewidth

A graph class $\mathcal{C}$ has *locally bounded treewidth* if there exists a function $g(r)$ such that for every graph $G \in \mathcal{C}$, every $v \in V(G)$ and every $r \in \mathbb{N}$ the neighborhood $G[N_r(v)]$ has treewidth at most $g(r)$.

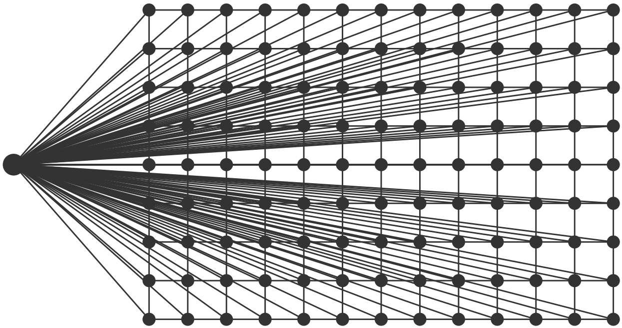
Which of these classes have locally bounded treewidth?



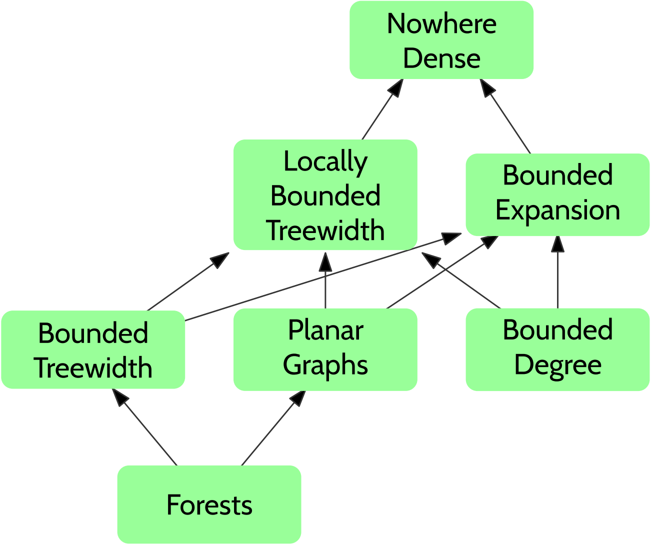
Locally Bounded Treewidth

A graph class $\mathcal{C}$ has *locally bounded treewidth* if there exists a function $g(r)$ such that for every graph $G \in \mathcal{C}$, every $v \in V(G)$ and every $r \in \mathbb{N}$ the neighborhood $G[N_r(v)]$ has treewidth at most $g(r)$.

Which of these classes have locally bounded treewidth?



Relationship to Other Classes



Relationship to Other Classes

Locally bounded treewidth generalizes

- planar graphs,
- bounded degree,
- bounded treewidth.

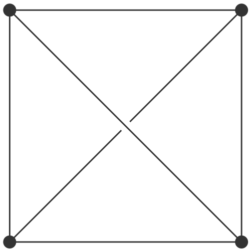
Theorem (Bodlaender)

The treewidth of a planar graph with radius r is at most $3r - 1$.

The size of a graph with with radius r and maximal degree d is at most d^{r+1}.

Subdivisions

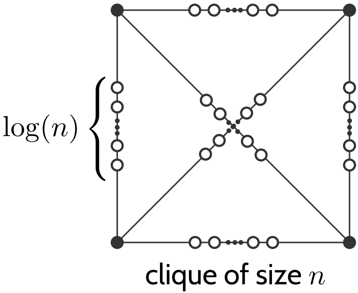
Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?



clique of size n

Subdivisions

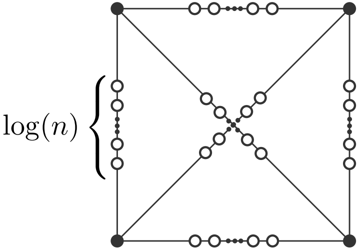
Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?



Subdivisions

Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?

Yes! Proof:

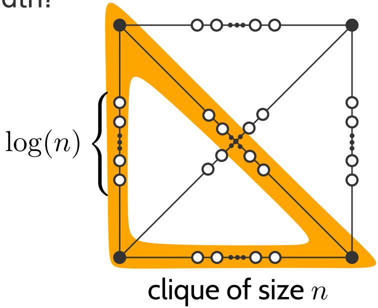


clique of size n

Subdivisions

Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?

Yes! Proof:

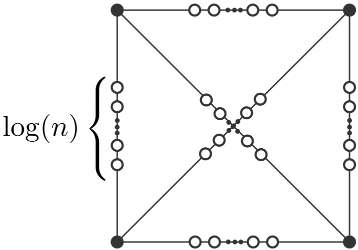


- Every circle has length at least $3\log(n)$.

Subdivisions

Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?

Yes! Proof:



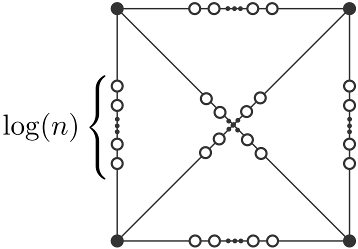
clique of size n

- Every circle has length at least $3\log(n)$.
- If $r \leq \log(n)$ then $G[N_r(v)]$ has treewidth at most 1.

Subdivisions

Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?

Yes! Proof:



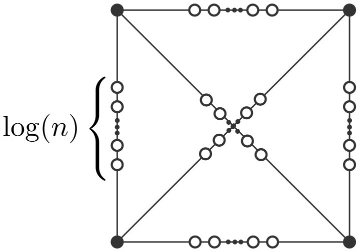
clique of size n

- Every circle has length at least $3\log(n)$.
- If $r \leq \log(n)$ then $G[N_r(v)]$ has treewidth at most 1.
- If $r > \log(n)$ then $G[N_r(v)]$ has treewidth at most $n \leq 2^r$.

Subdivisions

Does the class of $\log(n)$-subdivisions of n-cliques have locally bounded treewidth?

Yes! Proof:



clique of size n

- Every circle has length at least $3\log(n)$.
- If $r \leq \log(n)$ then $G[N_r(v)]$ has treewidth at most 1.
- If $r > \log(n)$ then $G[N_r(v)]$ has treewidth at most $n \leq 2^r$.
- For every r, $G[N_r(v)]$ has treewidth at most 2^r.

A More General First-Order Meta-Theorem

Theorem (Frick, Grohe 2001)

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

A More General First-Order Meta-Theorem

Theorem (Frick, Grohe 2001)

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

We prove the following slightly weaker statement.

Theorem (Frick, Grohe 2001)

Let $\mathcal{C}$ be a graph class with locally bounded treewidth. 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 $f(|\varphi|)n^2$.

Proof

Theorem (Frick, Grohe 2001)

Let $\mathcal{C}$ be a graph class with locally bounded treewidth. 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 $f(|\varphi|)n^2$.

Proof

Theorem (Frick, Grohe 2001)

Let $\mathcal{C}$ be a graph class with locally bounded treewidth. 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 $f(|\varphi|)n^2$.

Proof:

- By Gaifman's theorem, φ is equivalent of boolean combination of basic local sentences such as

$$\exists x_1 \dots \exists x_s \big(\bigwedge_{i \neq j} \mathrm{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \big).$$

  We show how to evaluate such sentences.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set R of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \big).$$

We compute the set R of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have treewidth at most $g(r)$.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set R of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have treewidth at most $g(r)$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $f'(|\omega|, g(r))n$ using Courcelle's theorem.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set R of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have treewidth at most $g(r)$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $f'(|\omega|, g(r))n$ using Courcelle's theorem.
- Doing it for all $v \in V(G)$ takes time $f'(|\omega|, g(r))n^2$.

Proof

Consider a basic local sentence

$$\exists x_1 \dots \exists x_s \Big(\bigwedge_{i \neq j} \mathsf{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set R of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have treewidth at most $g(r)$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $f'(|\omega|, g(r))n$ using Courcelle's theorem.
- Doing it for all $v \in V(G)$ takes time $f'(|\omega|, g(r))n^2$.

We have to find a red $2r$-scattered set of size s in G. This is equivalent to finding a red $2r$-scattered set of size s in $G[N_r(R)]$.

Proof

Consider a basic local sentence

$$\exists x_1 \ldots \exists x_s \Big(\bigwedge_{i \neq j} \text{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_i \omega(x_i) \Big).$$

We compute the set R of all vertices v with $G[N_r(v)] \models \omega(v)$ and color them red.

- Neighborhoods $G[N_r(v)]$ have treewidth at most $g(r)$.
- For $v \in V(G)$, check whether $G[N_r(v)] \models \omega(v)$ in time $f'(|\omega|, g(r))n$ using Courcelle's theorem.
- Doing it for all $v \in V(G)$ takes time $f'(|\omega|, g(r))n^2$.

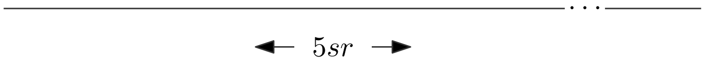
We have to find a red $2r$-scattered set of size s in G. This is equivalent to finding a red $2r$-scattered set of size s in $G[N_r(R)]$.

We do a case distinction on the diameter of $G[N_r(R)]$.

Proof

Assume $G[N_r(R)]$ has diameter at least $5sr$.

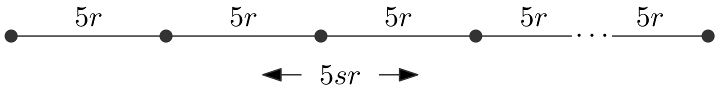
- Then there is a shortest path of length $5sr$.



Proof

Assume $G[N_r(R)]$ has diameter at least $5sr$.

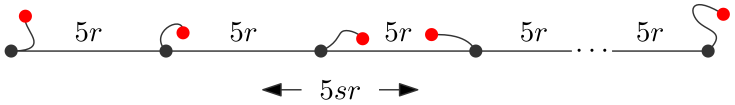
- Then there is a shortest path of length $5sr$.
- There are s vertices on it with distance $5r$ of each other.



Proof

Assume $G[N_r(R)]$ has diameter at least $5sr$.

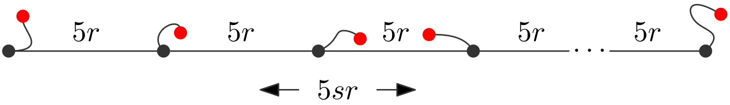
- Then there is a shortest path of length $5sr$.
- There are s vertices on it with distance $5r$ of each other.
- Each vertex on the path has distance $\leq r$ from a red vertex.



Proof

Assume $G[N_r(R)]$ has diameter at least $5sr$.

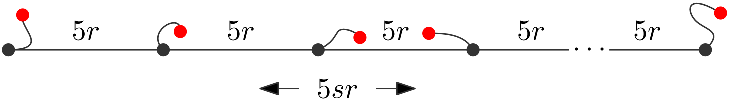
- Then there is a shortest path of length $5sr$.
- There are s vertices on it with distance $5r$ of each other.
- Each vertex on the path has distance $\leq r$ from a red vertex.
- These red vertices have pairwise distance $> 2r$.



Proof

Assume $G[N_r(R)]$ has diameter at least $5sr$.

- Then there is a shortest path of length $5sr$.
- There are s vertices on it with distance $5r$ of each other.
- Each vertex on the path has distance $\leq r$ from a red vertex.
- These red vertices have pairwise distance $> 2r$.
- $2r$-scattered set of size s exists!



Proof

Assume $G[N_r(R)]$ has diameter at most $5sr$. Then it has treewidth at most $g(5sr)$, where g is the treewidth bound of $\mathcal{C}$. Use Courcelle's theorem to decide if $G[N_r(R)]$ has red $2r$-scattered set.

Summary

We now have 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.