Monadically Stable Graph Classes

LoGAlg 2023, Warsaw, November 15

Jan Dreier, TU Wien

Main Question

First-Order Model Checking

Given a graph G and a first-order sentence φ, decide whether $G \models \varphi$.

Main Question

First-Order Model Checking

Given a graph G and a first-order sentence φ, decide whether $G \models \varphi$.

- Example: G has dominating set of size k if

$$G \models \exists x_1 \ldots \exists x_k \forall y \bigvee_i E(y, x_i) \vee y = x_i.$$

Main Question

First-Order Model Checking

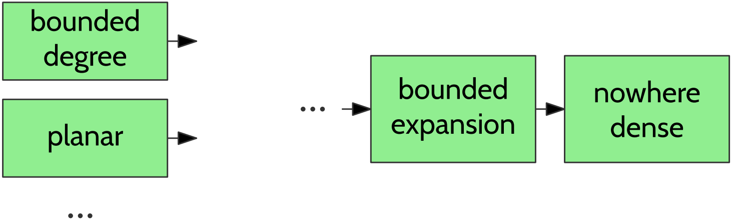
Given a graph G and a first-order sentence φ, decide whether $G \models \varphi$.

- Example: G has dominating set of size k if

$$G \models \exists x_1 \ldots \exists x_k \forall y \bigvee_i E(y, x_i) \vee y = x_i.$$

- Can be decided in $O(|G|^{|\varphi|})$.
- Question: On what graph classes is the problem fpt, i.e., solvable in time $f(|\varphi|) \cdot \text{poly}(|G|)$?

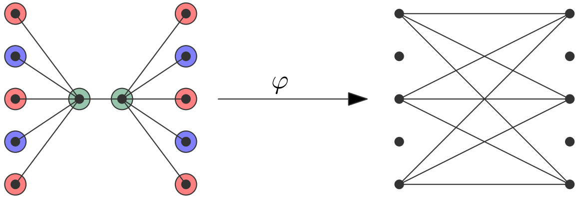
Tractable Classes



Bounded Degree Model Checking: Seese, 1996
Planar Model Checking: Flum, Grohe 2001
Bounded Expansion Model Checking: Dvořák, Král, Thomas, 2010
Nowhere Dense Model Checking: Grohe, Kreutzer, Siebertz, 2017

Transductions

φ-transduction: color vertices + apply φ + take induced subgraph

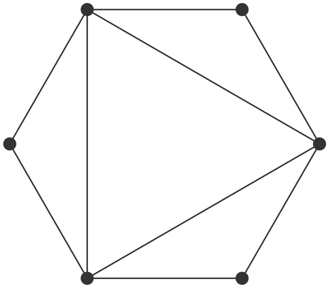


$\varphi(x, y) := \mathsf{Red}(x) \wedge \mathsf{Red}(y) \wedge \mathsf{dist}(x, y) = 3$

> A class $\mathcal{D}$ is a *transduction* of a class $\mathcal{C}$ if there exists φ such that every graph in $\mathcal{D}$ is a φ-transduction of some graph in $\mathcal{C}$.

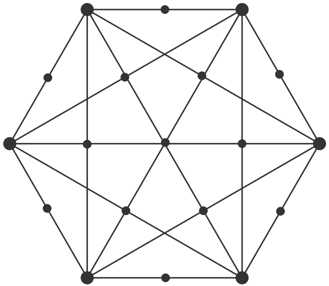
Transductions

The class of subdivided cliques transduces the class of all graphs.



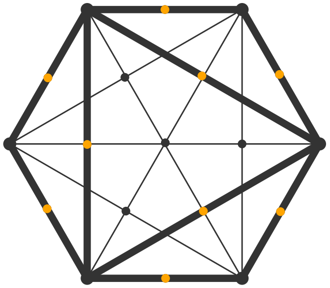
Transductions

The class of subdivided cliques transduces the class of all graphs.

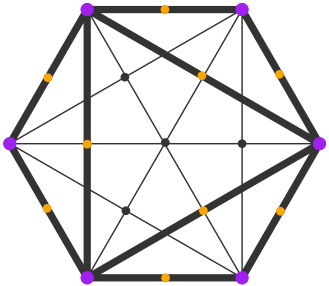


Transductions

The class of subdivided cliques transduces the class of all graphs.

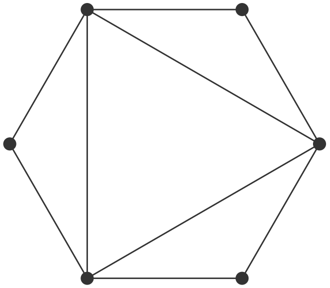


The class of subdivided cliques transduces the class of all graphs.



Transductions

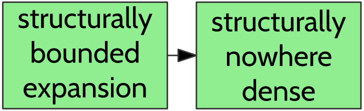
The class of subdivided cliques transduces the class of all graphs.



Tractable Classes

Gajarský, Kreutzer, Něsetřil, Ossona de Mendez, Pilipczuk, Siebertz, Toruńczyk, 2018.
Něsetřil, Ossona de Mendez, 2016

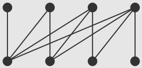
A class is *structurally nowhere dense*, if it is a transduction of a nowhere dense graph class.



Monadic Stability

Baldwin, Shelah, 1985

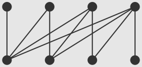
A class is *monadically stable*, if it does not transduce the class of all half-graphs.

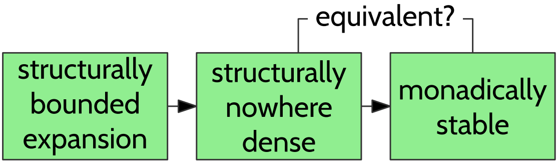


Tractable Classes

Baldwin, Shelah, 1985

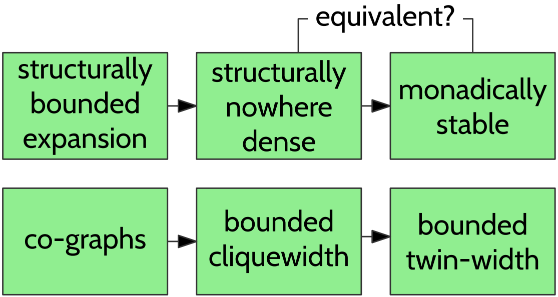
A class is *monadically stable*, if it does not transduce the class of all half-graphs.





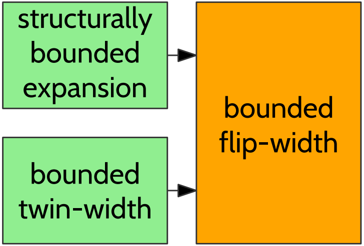
Monadically Stable Model Checking: Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Tractable Classes



Cliquewidth Model Checking: Courcelle, Makowsky, Rotics, 2000
Twin-Width Model Checking: Bonnet, Kim, Thomassé, Watrigant,2021,

Tractable Classes

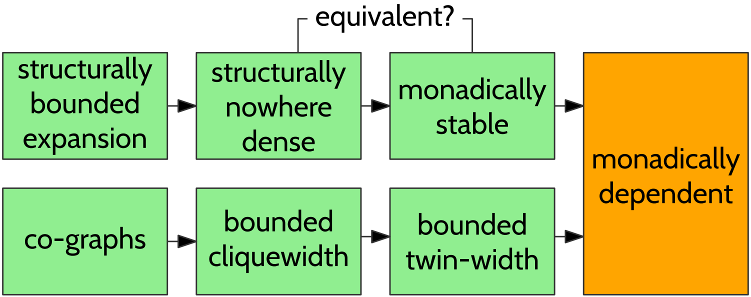


Flip-Width: Toruńczyk, 2023

Tractable Classes

Baldwin, Shelah, 1985

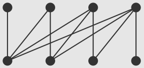
A class is *monadically dependent*, if it does not transduce the class of all graphs.



Tractable Classes

Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Let $\mathcal{C}$ be a hereditary graph class that does not contain arbitrariliy large *semi-induced* half-graphs.



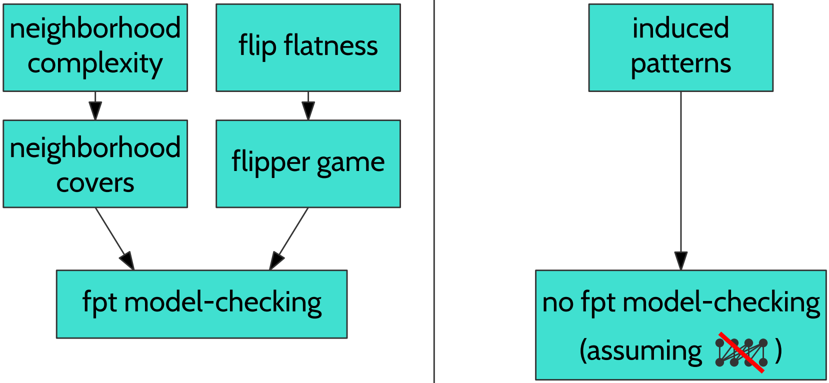
Model checking is fpt on $\mathcal{C}$

$$\Leftrightarrow$$

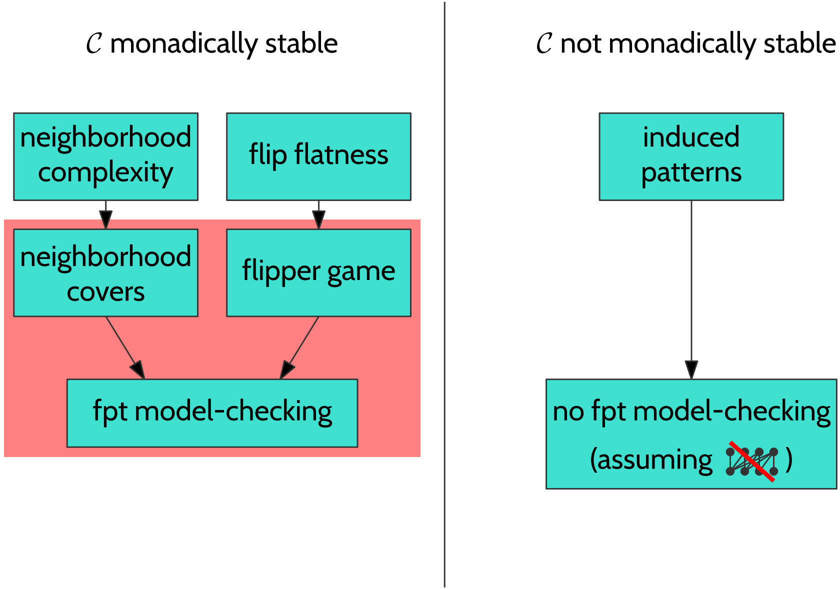
$\mathcal{C}$ is monadically stable

Outline

$\mathcal{C}$ monadically stable | $\mathcal{C}$ not monadically stable



Outline



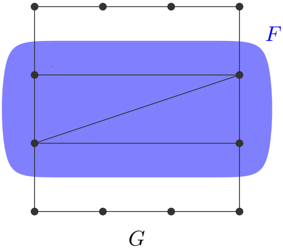
$\mathcal{C}$ monadically stable
neighborhood complexity
flip flatness
neighborhood covers
flipper game
fpt model-checking
$\mathcal{C}$ not monadically stable
induced patterns
no fpt model-checking
(assuming)

Flips

Denote by $G \oplus F$ the graph obtained from G by complementing edges between pairs of vertices from F.

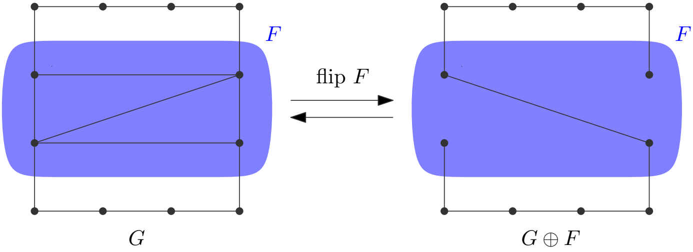
Flips

Denote by $G \oplus F$ the graph obtained from G by complementing edges between pairs of vertices from F.



Flips

Denote by $G \oplus F$ the graph obtained from G by complementing edges between pairs of vertices from F.



Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:

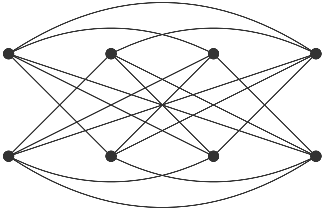
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



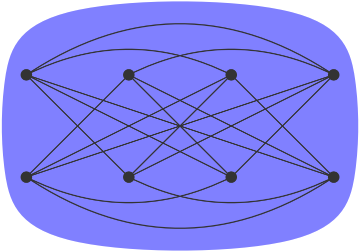
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



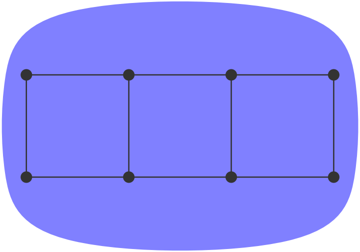
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



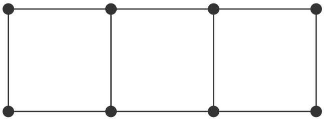
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



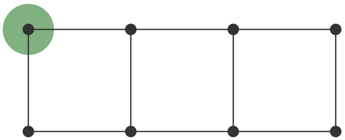
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



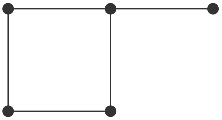
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



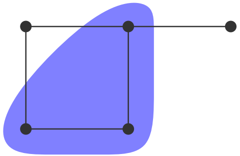
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



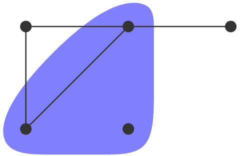
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



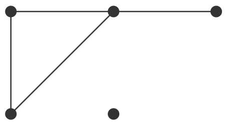
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



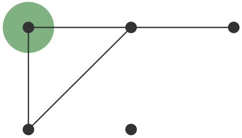
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



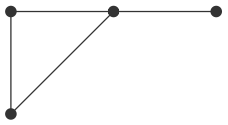
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



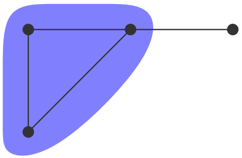
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



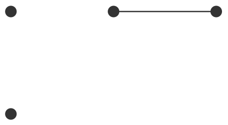
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



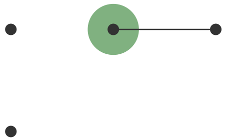
Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:



Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:

Flipper Game

The radius-r Flipper game is played on a graph G_1. In round i

1. Flipper/Pursuer chooses a flip set F
2. Evader chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

Example play of the radius-2 Flipper game:

Flipper Game and Monadic Stability

Gajarský, Mählmann, McCarty, Ohlmann, Pilipczuk, Przybyszewski, Siebertz, Sokołowski, Toruńczyk, 2023

A class of graphs $\mathcal{C}$ is monadically stable $\Leftrightarrow$

$\forall r \exists \ell$ such that Flipper wins the radius-r game on all graphs from $\mathcal{C}$ in ℓ rounds.

Flipper Game and Monadic Stability

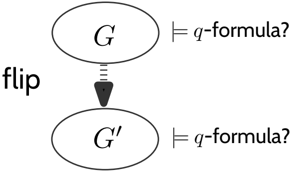
Gajarský, Mählmann, McCarty, Ohlmann, Pilipczuk, Przybyszewski, Siebertz, Sokołowski, Toruńczyk, 2023

A class of graphs $\mathcal{C}$ is monadically stable $\Leftrightarrow$

$\forall r \exists \ell$ such that Flipper wins the radius-r game on all graphs from $\mathcal{C}$ in ℓ rounds.

Moreover, Flipper's moves can be computed in time $O(n^2)$.

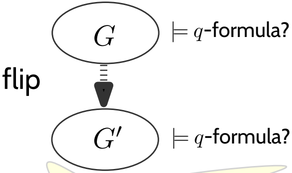
Guiding the Recursion with Flipper Games



round 1

Algorithm from: Dreier, Mählmann, Siebertz, 2022

Guiding the Recursion with Flipper Games

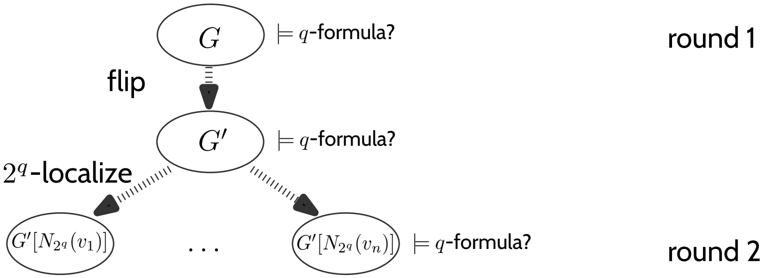


round 1

Update q-formula by replacing each edge relation:

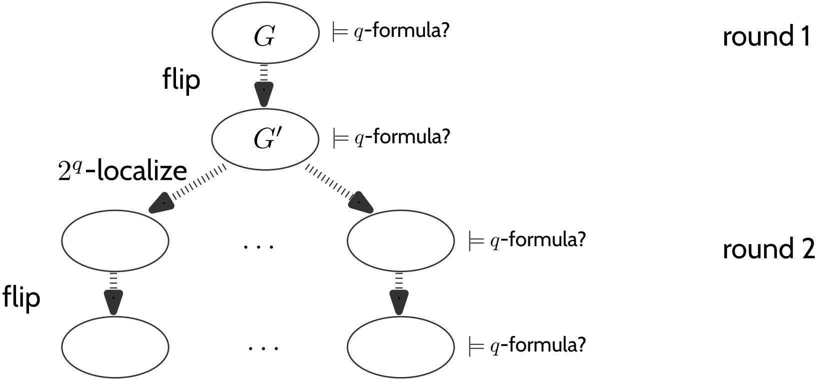
$$G \models E(x,y) \iff G' \models E(x,y) \oplus (x \in F \wedge y \in F)$$

Guiding the Recursion with Flipper Games



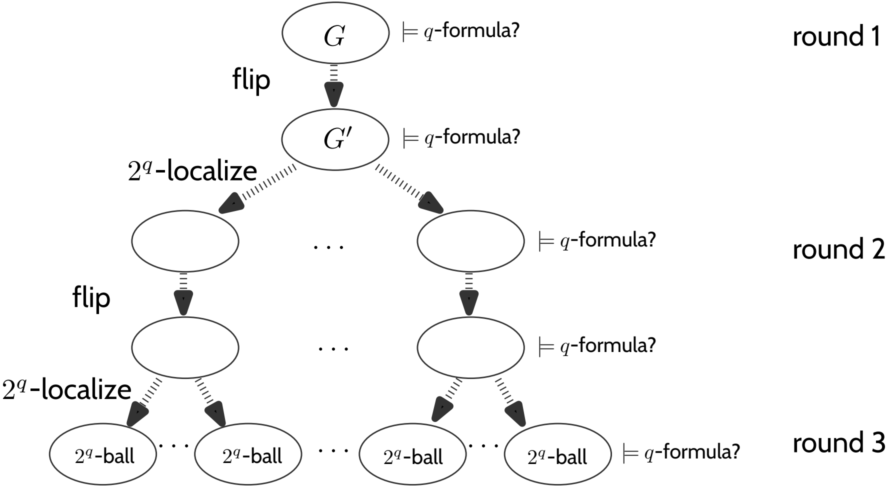
Algorithm from: Dreier, Mählmann, Siebertz, 2022

Guiding the Recursion with Flipper Games



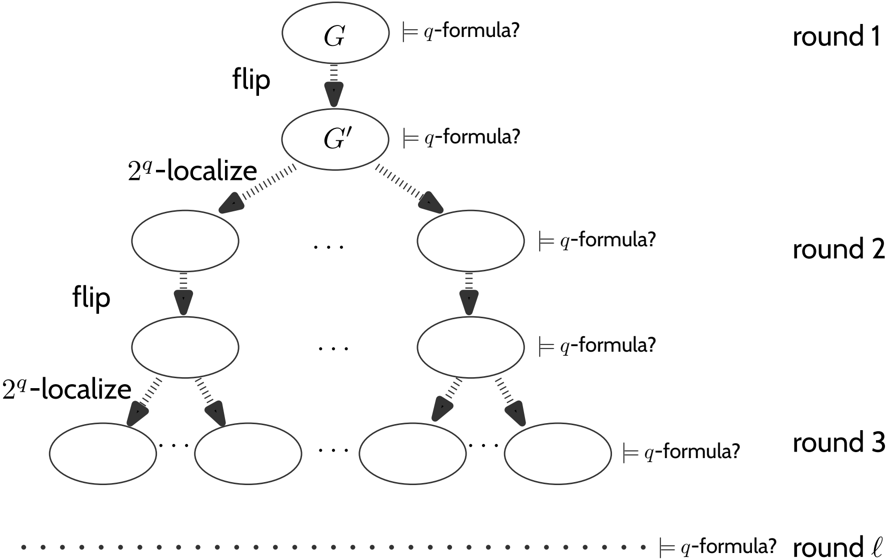
Algorithm from: Dreier, Mählmann, Siebertz, 2022

Guiding the Recursion with Flipper Games



Algorithm from: Dreier, Mählmann, Siebertz, 2022

Guiding the Recursion with Flipper Games



Algorithm from: Dreier, Mählmann, Siebertz, 2022

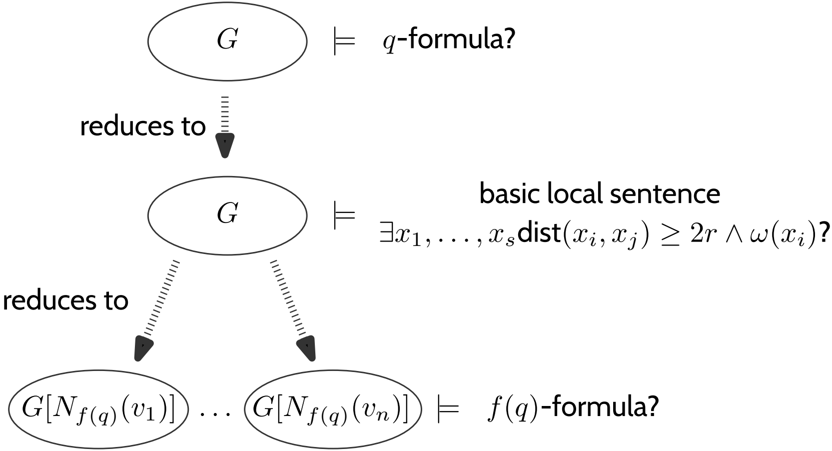
Gaifman–Approach

$G \models$ q-formula?

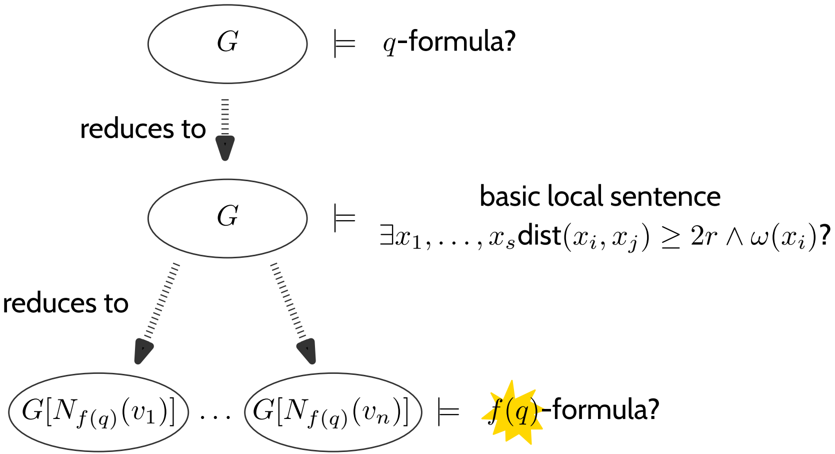
reduces to

$G \models$ basic local sentence $\exists x_1, \ldots, x_s \mathsf{dist}(x_i, x_j) \geq 2r \wedge \omega(x_i)$?

Gaifman–Approach



Gaifman–Approach



Local Types

Reduce model checking to the problem of deciding whether two vertices have the same q-type.

Local Types

Reduce model checking to the problem of deciding whether two vertices have the same q-type.

Gajarský, Gorsky, Kreutzer, 2020 Toruńczyk, 2022 Dreier, Mählmann, Siebertz, 2022

Let G be a graph and a, b be two vertices with distance more than 2^q and

$$q\text{-type}(G[N_{2^q}(a)], a) = q\text{-type}(G[N_{2^q}(b)], b).$$

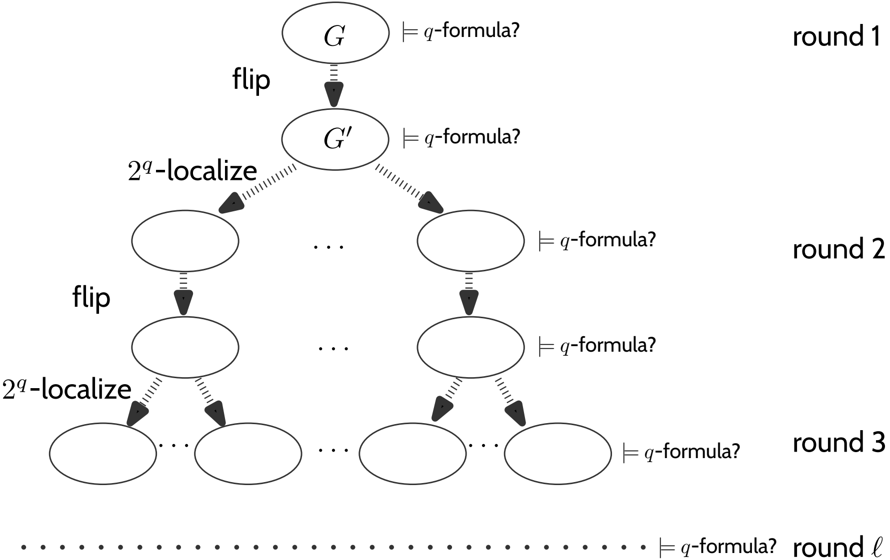
Then

$$q\text{-type}(G, a) = q\text{-type}(G, b).$$

Guiding the Recursion with Flipper Games

There is just one more problem...

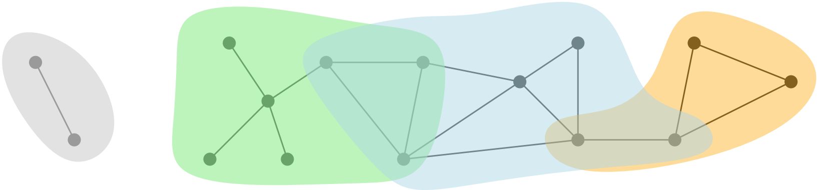
Guiding the Recursion with Flipper Games



Neighborhood Covers

Technique introduced by: Grohe, Kreutzer, Siebertz, 2017

We say an r-ball is a subgraph with radius r. An r-*neighborhood cover* with degree Δ in a graph G is a collection of sets $C_1, \dots, C_l \subseteq V(G)$ such that



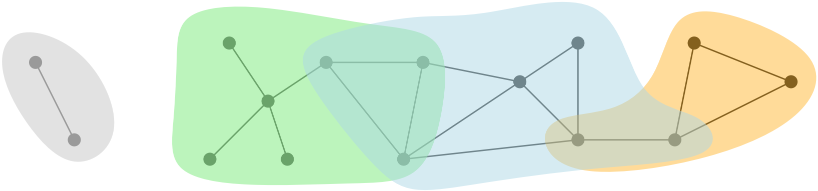
1-neigbhorhood cover with degree 2

Neighborhood Covers

Technique introduced by: Grohe, Kreutzer, Siebertz, 2017

We say an r-ball is a subgraph with radius r. An r-*neighborhood cover* with degree Δ in a graph G is a collection of sets $C_1, \ldots, C_l \subseteq V(G)$ such that

- every r-ball of G is contained in some C_i,



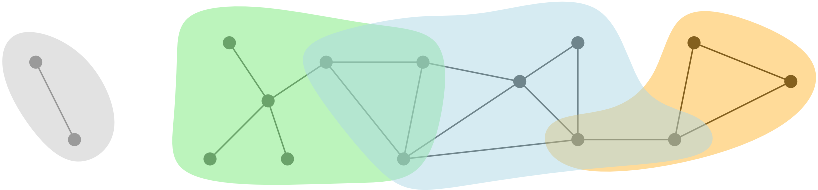
1-neigbhorhood cover with degree 2

Neighborhood Covers

Technique introduced by: Grohe, Kreutzer, Siebertz, 2017

We say an r-ball is a subgraph with radius r. An r-*neighborhood cover* with degree Δ in a graph G is a collection of sets $C_1, \ldots, C_l \subseteq V(G)$ such that

- every r-ball of G is contained in some C_i,
- every C_i is contained in some $4r$-ball of G,



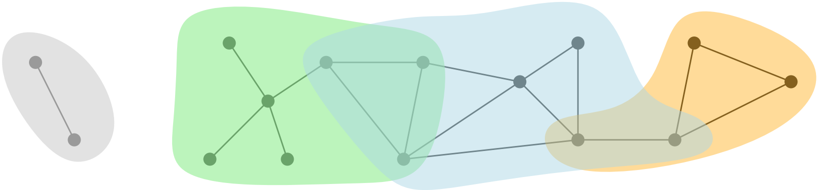
1-neigbhorhood cover with degree 2

Neighborhood Covers

Technique introduced by: Grohe, Kreutzer, Siebertz, 2017

We say an r-ball is a subgraph with radius r. An r-*neighborhood cover* with degree Δ in a graph G is a collection of sets $C_1, \dots, C_l \subseteq V(G)$ such that

- every r-ball of G is contained in some C_i,
- every C_i is contained in some $4r$-ball of G,
- every vertex of G is contained in at most Δ many C_i.



1-neigbhorhood cover with degree 2

Neighborhood Covers

Technique introduced by: Grohe, Kreutzer, Siebertz, 2017

We say an r-ball is a subgraph with radius r. An r*-neighborhood cover* with degree Δ in a graph G is a collection of sets $C_1, \dots, C_l \subseteq V(G)$ such that

- every r-ball of G is contained in some C_i,
- every C_i is contained in some $4r$-ball of G,
- every vertex of G is contained in at most Δ many C_i.

Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Let $\mathcal{C}$ be a monadically stable graph class. Every $G \in \mathcal{C}$ has an r-neighborhood cover with degree $O_{\mathcal{C},\varepsilon,r}(|G|^{\varepsilon})$ for all $\varepsilon > 0$, $r \in \mathbb{N}$.

Neighborhood Covers

Technique introduced by: Grohe, Kreutzer, Siebertz, 2017

We say an r-ball is a subgraph with radius r. An r-*neighborhood cover* with degree Δ in a graph G is a collection of sets $C_1, \ldots, C_l \subseteq V(G)$ such that

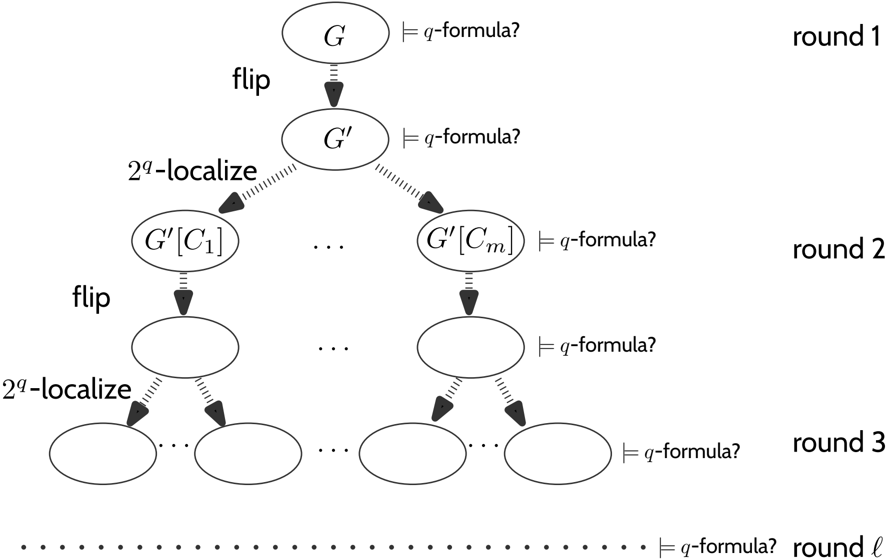
- every r-ball of G is contained in some C_i,
- every C_i is contained in some $4r$-ball of G,
- every vertex of G is contained in at most Δ many C_i.

Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Let $\mathcal{C}$ be a monadically stable graph class. Every $G \in \mathcal{C}$ has an r-neighborhood cover with degree $O_{\mathcal{C},\varepsilon,r}(|G|^\varepsilon)$ for all $\varepsilon > 0$, $r \in \mathbb{N}$.

Then in particular, $\sum_{i=1}^{l} |C_i| \leq n \cdot O_{\mathcal{C},\varepsilon,r}(|G|^\varepsilon)$.

Bounding the Size using Neighborhood Covers



Model Checking Summary

In summary, we used three ingredients.

Model Checking Summary

In summary, we used three ingredients.

- A quantifier-rank preserving localization procedure.

Model Checking Summary

In summary, we used three ingredients.

- A quantifier-rank preserving localization procedure.
- The Flipper game bounds the depth of the recursion tree.

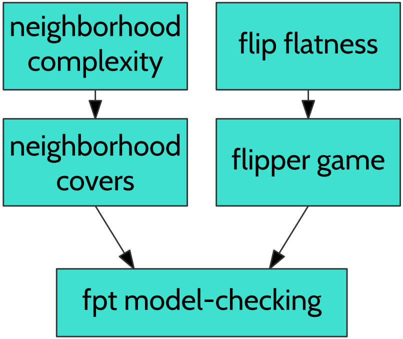
Model Checking Summary

In summary, we used three ingredients.

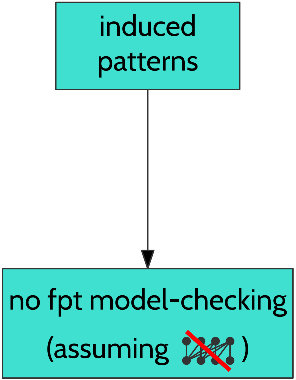
- A quantifier-rank preserving localization procedure.
- The Flipper game bounds the depth of the recursion tree.
- The neighborhood covers bound the size of the recursion tree.

Outline

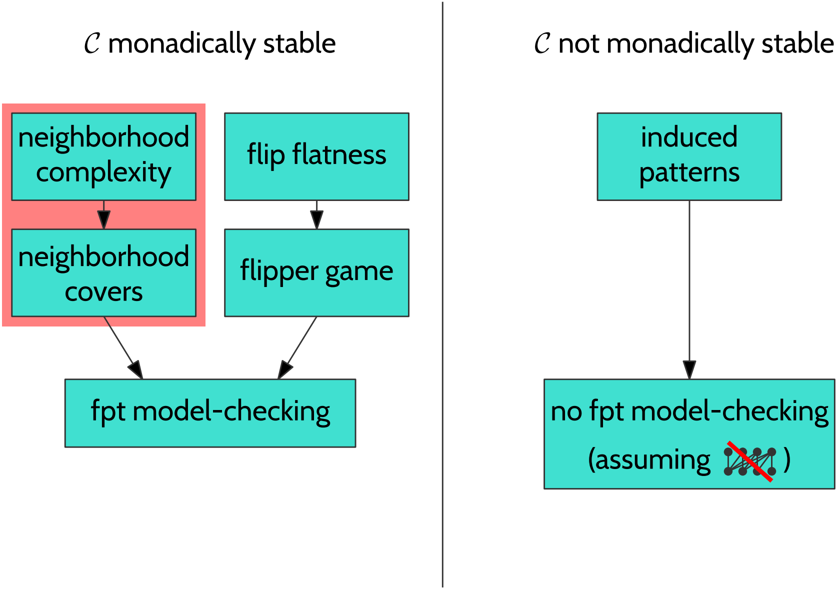
$\mathcal{C}$ monadically stable



$\mathcal{C}$ not monadically stable

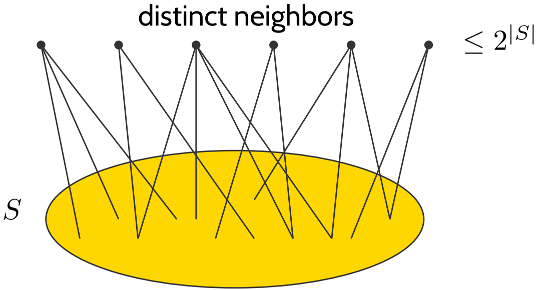


Outline



$\mathcal{C}$ monadically stable
$\mathcal{C}$ not monadically stable
neighborhood complexity
flip flatness
neighborhood covers
flipper game
fpt model-checking
induced patterns
no fpt model-checking
(assuming)

Neighborhood Complexity

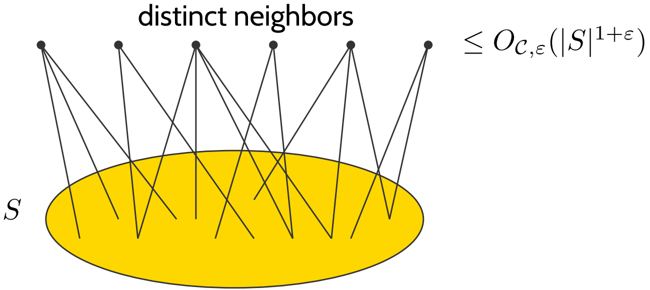


Neighborhood Complexity

Eickmeyer, Giannopoulou, Kreutzer, Kwon, Pilipczuk, Rabinovich, Siebertz, 2016

Let $\mathcal{C}$ be a nowhere dense graph class. For all $\varepsilon > 0$, $G \in \mathcal{C}$, $S \subseteq V(G)$,

$$|\{N(v) \cap S \mid v \in V(G)\}| \leq O_{\mathcal{C},\varepsilon}(|S|^{1+\varepsilon}).$$

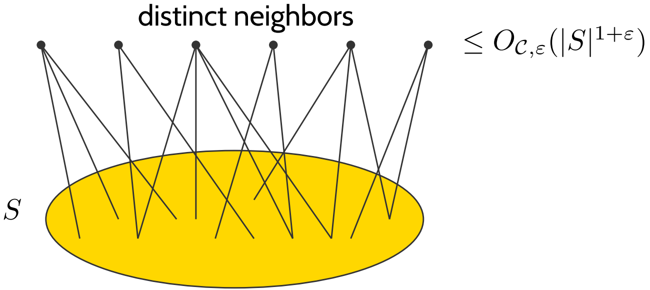


Neighborhood Complexity

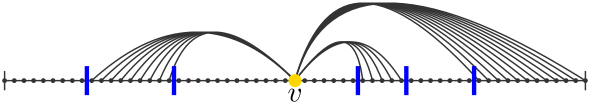
Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Let $\mathcal{C}$ be a monadically stable graph class. For all $\varepsilon > 0$, $G \in \mathcal{C}$, $S \subseteq V(G)$,

$$|\{N(v) \cap S \mid v \in V(G)\}| \leq O_{\mathcal{C},\varepsilon}(|S|^{1+\varepsilon}).$$

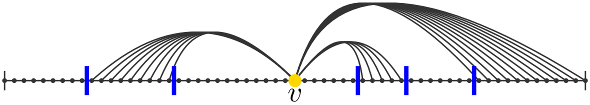


Welzl Orders



Welzl-order where v has 5 alternations

Welzl Orders



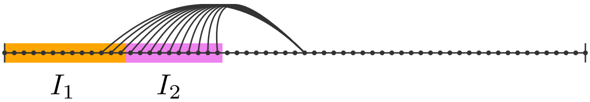
Welzl-order where v has 5 alternations

Corollary, Welzl, 1988

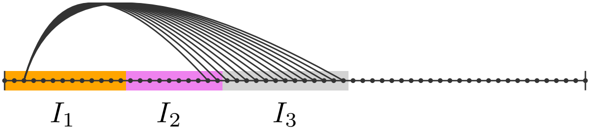
If a graph class $\mathcal{C}$ is monadically stable, then for all $\varepsilon > 0$ exists $c \in \mathbb{N}$ such that all $G \in \mathcal{C}$ admit Welzl orders where each vertex has $c \cdot |G|^{1+\varepsilon}$ alternations.

Building 1-Neighborhood Covers

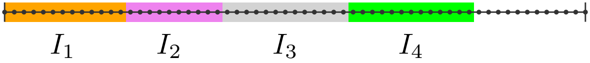
Building 1-Neighborhood Covers



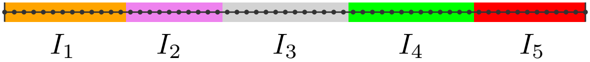
Building 1-Neighborhood Covers



Building 1-Neighborhood Covers



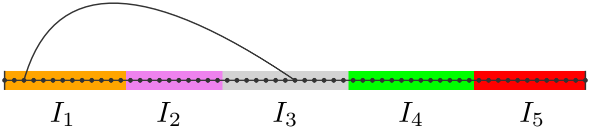
Building 1-Neighborhood Covers



Building 1-Neighborhood Covers

Let $N(I_1), N(I_2), \ldots$ be the clusters of the 1-neighborhood cover.

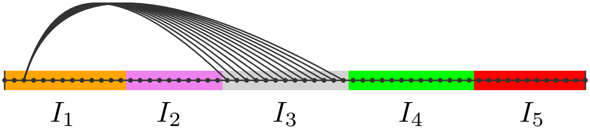
Building 1-Neighborhood Covers



Let $N(I_1), N(I_2), \ldots$ be the clusters of the 1-neighborhood cover.

Every vertex in $N(I_i)$ has alternation point in I_i.

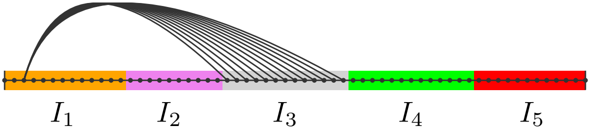
Building 1-Neighborhood Covers



Let $N(I_1), N(I_2), \ldots$ be the clusters of the 1-neighborhood cover.

Every vertex in $N(I_i)$ has alternation point in I_i.

Building 1-Neighborhood Covers

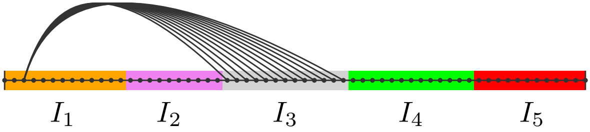


Let $N(I_1), N(I_2), \ldots$ be the clusters of the 1-neighborhood cover.

Every vertex in $N(I_i)$ has alternation point in I_i.

$\Rightarrow$ Every vertex is in $O(n^\varepsilon)$ clusters.

Building 1-Neighborhood Covers



Let $N(I_1), N(I_2), \ldots$ be the clusters of the 1-neighborhood cover.

Every vertex in $N(I_i)$ has alternation point in I_i.

$\Rightarrow$ Every vertex is in $O(n^\varepsilon)$ clusters.

Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Let $\mathcal{C}$ be a monadically stable graph class. Every $G \in \mathcal{C}$ has a 1-neighborhood cover with degree $O_{\varepsilon,\mathcal{C}}(|G|^\varepsilon)$ for all $\varepsilon > 0$.

Outline

$\mathcal{C}$ monadically stable

neighborhood complexity → neighborhood covers

flip flatness → flipper game

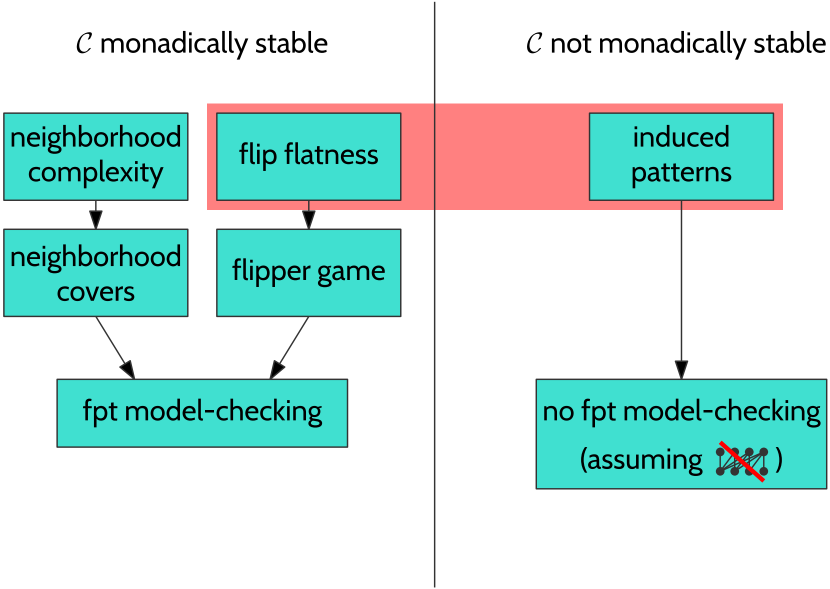
neighborhood covers → fpt model-checking

flipper game → fpt model-checking

$\mathcal{C}$ not monadically stable

induced patterns → no fpt model-checking (assuming )

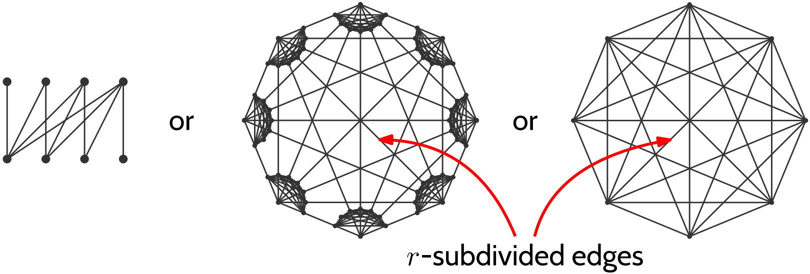
Outline


$\mathcal{C}$ monadically stable
$\mathcal{C}$ not monadically stable
neighborhood complexity
flip flatness
induced patterns
neighborhood covers
flipper game
fpt model-checking
no fpt model-checking
(assuming)

Forbidden Patterns

Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

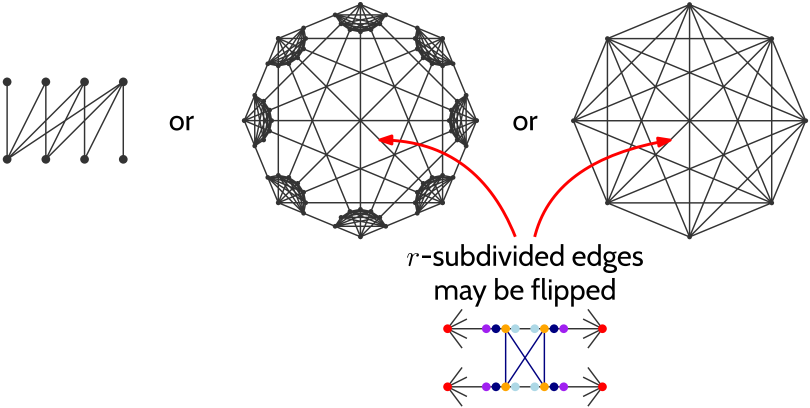
A class $\mathcal{C}$ is monadically stable if and only if for every r, these three types of induced subgraphs appear only up to a certain size.



Forbidden Patterns

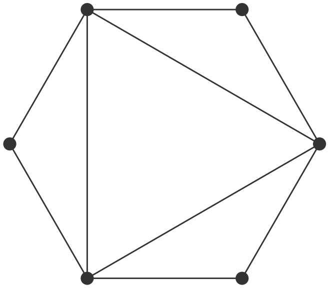
Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

A class $\mathcal{C}$ is monadically stable if and only if for every r, these three types of induced subgraphs appear only up to a certain size.



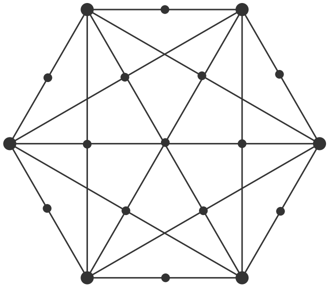
Transductions

The right two patterns can be used to show hardness of model checking.

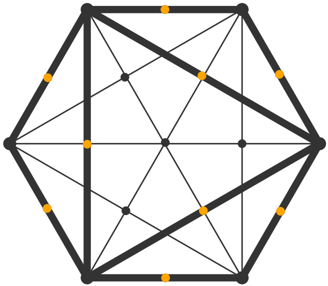


Transductions

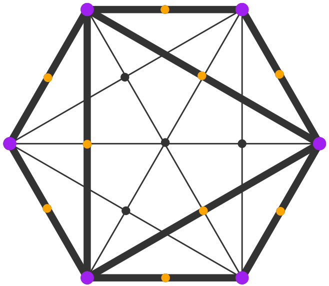
The right two patterns can be used to show hardness of model checking.



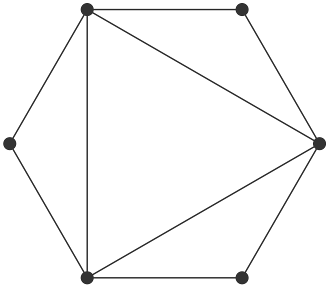
The right two patterns can be used to show hardness of model checking.



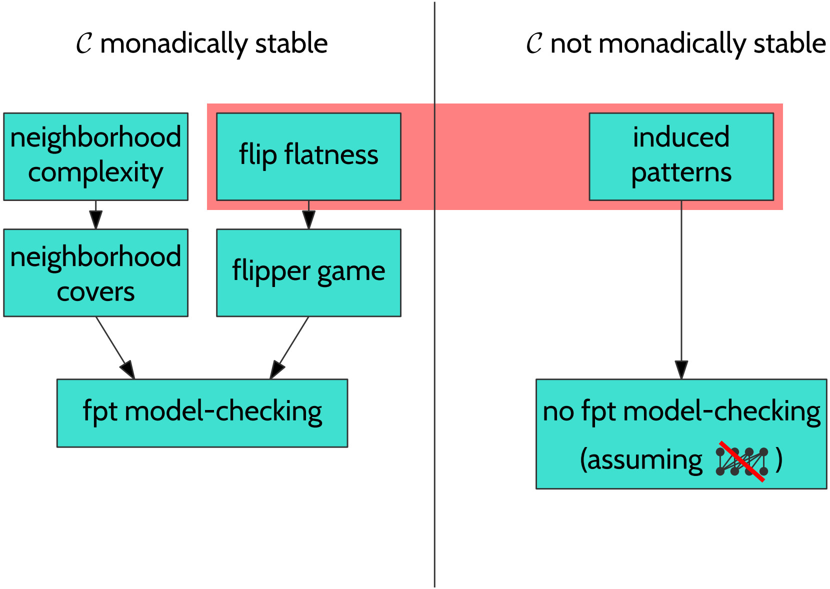
The right two patterns can be used to show hardness of model checking.



The right two patterns can be used to show hardness of model checking.



Outline

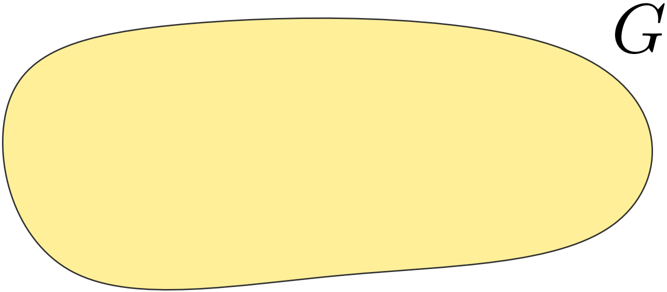


$\mathcal{C}$ monadically stable
$\mathcal{C}$ not monadically stable
neighborhood complexity
flip flatness
induced patterns
neighborhood covers
flipper game
fpt model-checking
no fpt model-checking
(assuming)

Ramsey

Ramsey's Theorem

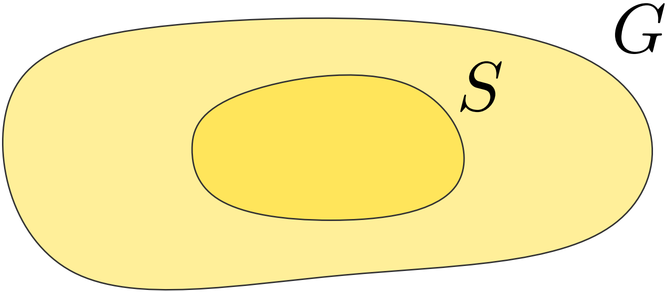
In every graph G



Ramsey

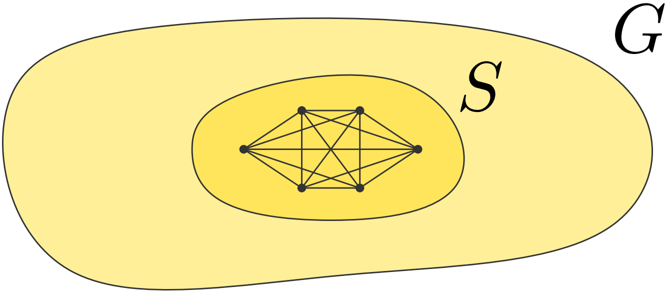
Ramsey's Theorem

In every graph G we can find $S \subseteq V(G)$, $|S| = U(|G|)$



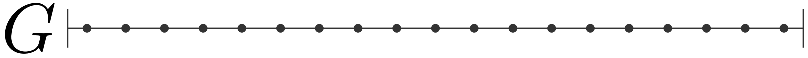
Ramsey's Theorem

In every graph G we can find $S \subseteq V(G)$, $|S| = U(|G|)$ such that S forms an independent set or clique.



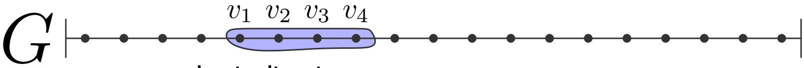
Can we force even more structure onto S?

Fix $q \in \mathbb{N}$ and order the vertices of G.



Can we force even more structure onto S?

Fix $q \in \mathbb{N}$ and order the vertices of G.

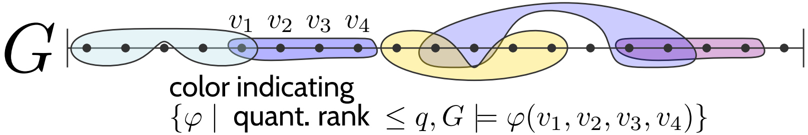


color indicating
$\{\varphi \mid$ quant. rank $\leq q, G \models \varphi(v_1, v_2, v_3, v_4)\}$

Ramsey

Can we force even more structure onto S?

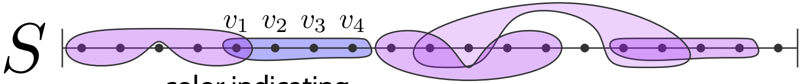
Fix $q \in \mathbb{N}$ and order the vertices of G.



Ramsey

Can we force even more structure onto S?

Fix $q \in \mathbb{N}$ and order the vertices of G.



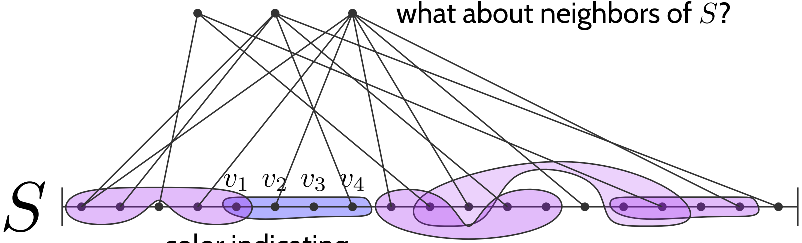
color indicating
$\{\varphi \mid$ quant. rank $\leq q, G \models \varphi(v_1, v_2, v_3, v_4)\}$

all tuples have same color

Can we force even more structure onto S?

Fix $q \in \mathbb{N}$ and order the vertices of G.

what about neighbors of S?



color indicating
$\{\varphi \mid$ quant. rank $\leq q, G \models \varphi(v_1, v_2, v_3, v_4)\}$

all tuples have same color

Can we force even more structure onto S?

Fix $q \in \mathbb{N}$ and order the vertices of G.

what about neighbors of S?

Question

Can we use monadic stability to force structure not only onto S, but also its neighbors?

Motivation: Uniform Quasi-Wideness

all tuples have same color

Flip Flatness

Dreier, Mählmann, Toruńczyk, Siebertz 2023

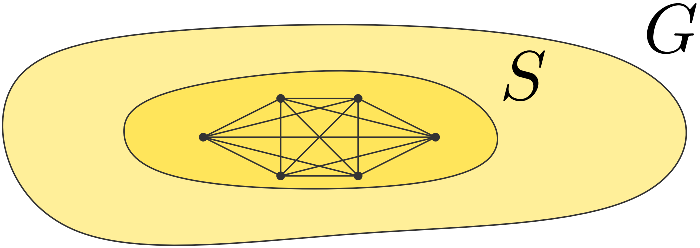
Let $\mathcal{C}$ be monadically stable, $r \in \mathbb{N}$. There exists $c \in \mathbb{N}$ with following property.

Flip Flatness

Dreier, Mählmann, Toruńczyk, Siebertz 2023

Let $\mathcal{C}$ be monadically stable, $r \in \mathbb{N}$. There exists $c \in \mathbb{N}$ with following property.

In every $G \in \mathcal{C}$ we find $S \subseteq V(G)$, $|S| = U_r(|G|)$, such that ~~S forms an independent set of clique~~



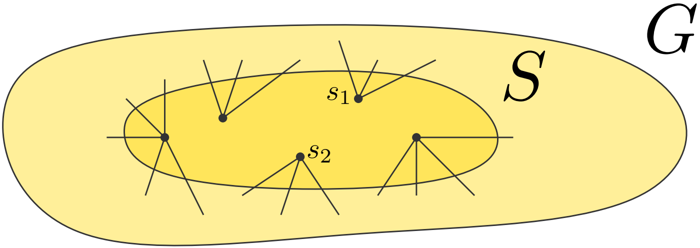
Flip Flatness

Dreier, Mählmann, Toruńczyk, Siebertz 2023

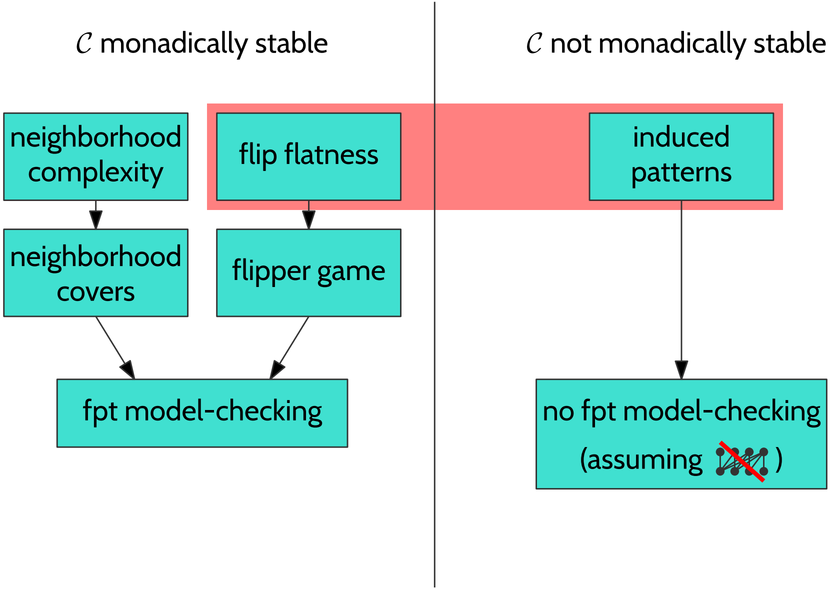
Let $\mathcal{C}$ be monadically stable, $r \in \mathbb{N}$. There exists $c \in \mathbb{N}$ with following property.

In every $G \in \mathcal{C}$ we find $S \subseteq V(G)$, $|S| = U_r(|G|)$, such that ~~S forms an independent set of clique~~

after performing c flips, $\forall s_1, s_2 \in S \;\; N_r(s_1) \cap N_r(s_2) = \emptyset$.



Outline


$\mathcal{C}$ monadically stable
$\mathcal{C}$ not monadically stable
neighborhood complexity
flip flatness
induced patterns
neighborhood covers
flipper game
fpt model-checking
no fpt model-checking
(assuming)

Goal

Let $\mathcal{C}$ be *any* graph class.

Goal

Let $\mathcal{C}$ be *any* graph class.

Either patterns of arbitrary size appear (not monadically stable),

Goal

Let $\mathcal{C}$ be *any* graph class.

Either patterns of arbitrary size appear (not monadically stable), or there is no half-graph, 1-subdivided clique, or its complement of size $t \in \mathbb{N}$.

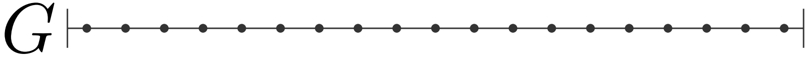
Goal

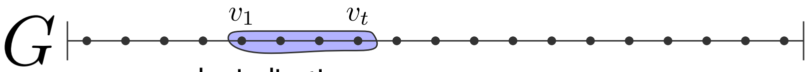
Let $\mathcal{C}$ be *any* graph class.

Either patterns of arbitrary size appear (not monadically stable), or there is no half-graph, 1-subdivided clique, or its complement of size $t \in \mathbb{N}$.

Use this to show radius-1 flip-flatness of $\mathcal{C}$.

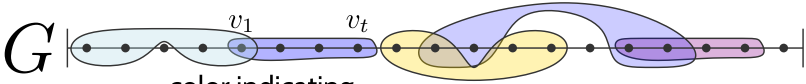
Ramsey





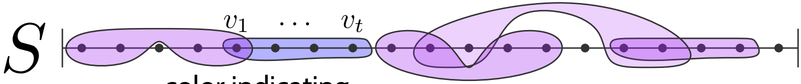
color indicating
$\{\varphi \mid$ quant. rank $\leq q, G \models \varphi(v_1, \ldots v_t)\}$

Ramsey



color indicating
$\{\varphi \mid$ quant. rank $\leq q, G \models \varphi(v_1, \ldots v_t)\}$

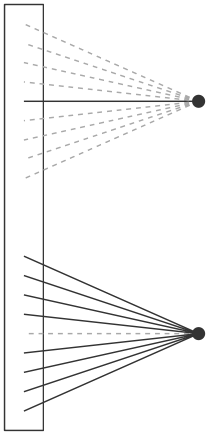
Ramsey



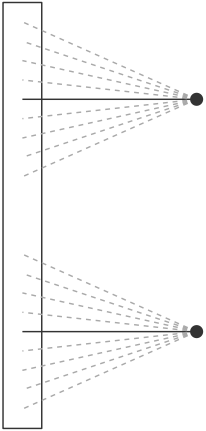
color indicating
$\{\varphi \mid$ quant. rank $\leq q, G \models \varphi(v_1, \ldots v_t)\}$

all tuples have same color

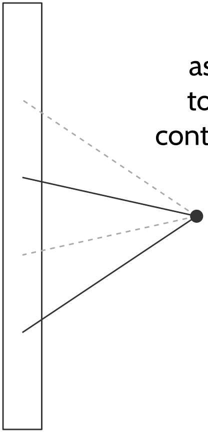
Construction



Construction

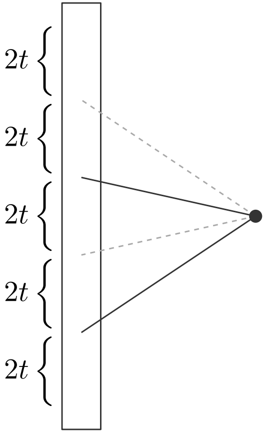


Construction

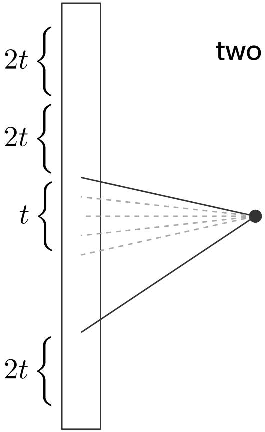


assume
towards
contradiction

Construction

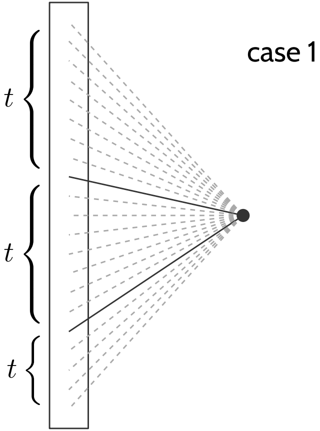


Construction

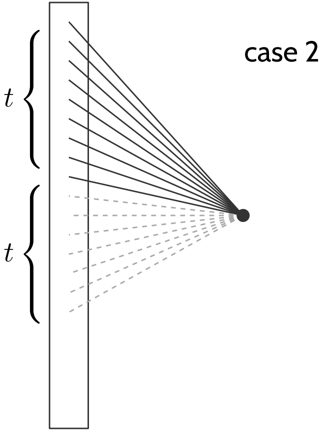


two cases

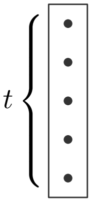
Construction

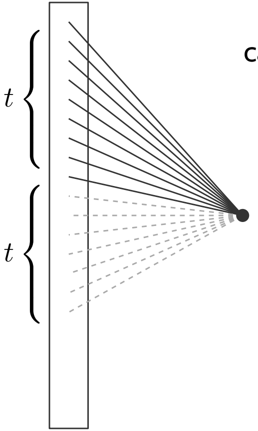


Construction

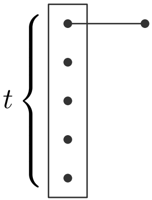


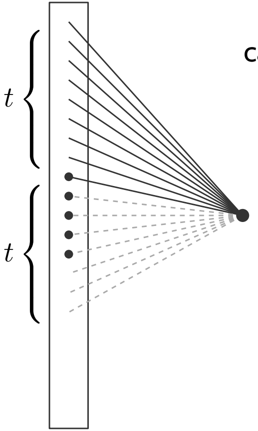
Construction





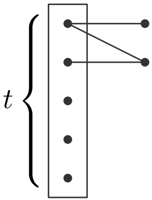
Construction

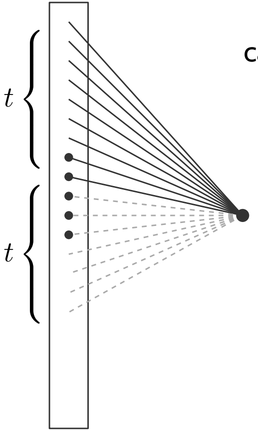




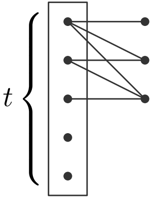
case 2

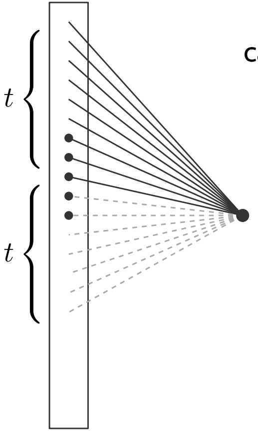
Construction





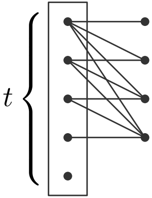
Construction

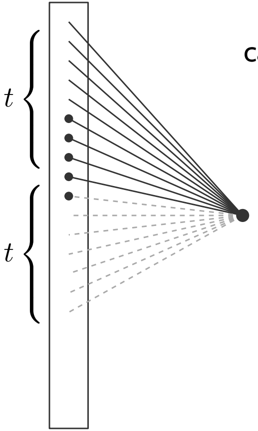




case 2

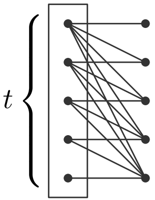
Construction

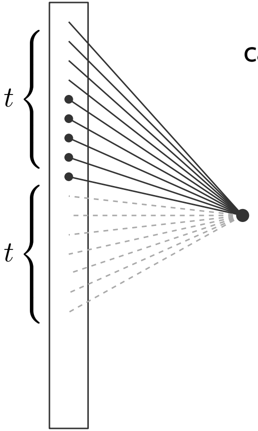




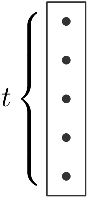
case 2

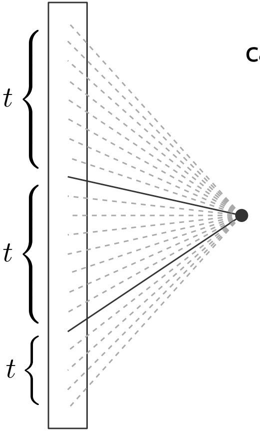
Construction



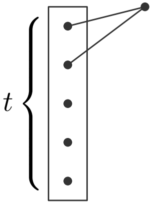


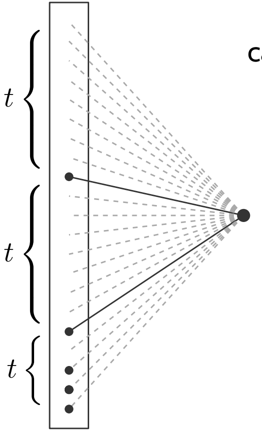
Construction





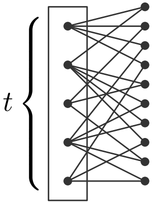
Construction

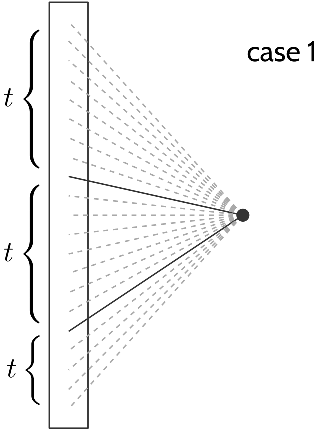




case 1

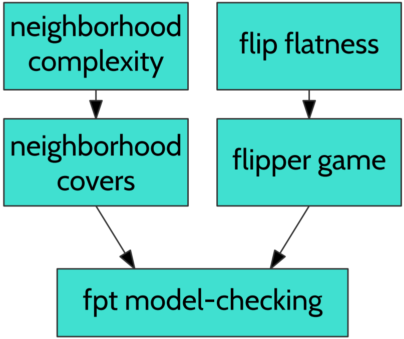
Construction



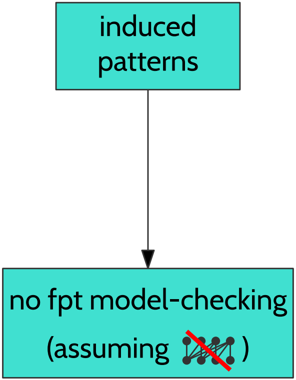


Outline

$\mathcal{C}$ monadically stable



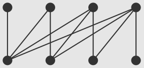
$\mathcal{C}$ not monadically stable



Summary

Dreier, Eleftheriadis, Mählmann, McCarty, Pilipczuk, Toruńczyk, 2023

Let $\mathcal{C}$ be a hereditary graph class that does not contain arbitrariliy large *semi-induced* half-graphs.



Model checking is fpt on $\mathcal{C}$

$\Leftrightarrow$

$\mathcal{C}$ is monadically stable

The End