Flipper Games for Monadically Stable Graph Classes

Jakub Gajarský, Nikolas Mählmann, Rose McCarty,
Pierre Ohlmann, Michał Pilipczuk, Wojciech Przybyszewski,
Sebastian Siebertz, Marek Sokołowski, Szymon Toruńczyk

ICALP 2023

Nowhere Dense Classes of Graphs

Definition [Něsetřil, Ossona de Mendez, 2011]

A class $\mathcal{C}$ is *nowhere dense*, if for every r there exists k such $\mathcal{C}$ that does not contain the r-subdivided clique of size k as a subgraph.

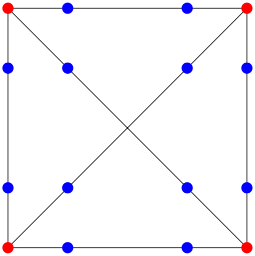


Figure: The 2-subdivided K_4.

Nowhere Dense Classes of Graphs

Definition [Něsetřil, Ossona de Mendez, 2011]

A class $\mathcal{C}$ is *nowhere dense*, if for every r there exists k such $\mathcal{C}$ that does not contain the r-subdivided clique of size k as a subgraph.

Generalizes many notions of sparsity such as: bounded degree, bounded treewidth, planarity, excluding a minor, ...

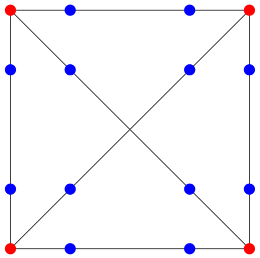


Figure: The 2-subdivided K_4.

Nowhere Dense Classes of Graphs

Definition [Něsetřil, Ossona de Mendez, 2011]

A class $\mathcal{C}$ is *nowhere dense*, if for every r there exists k such $\mathcal{C}$ that does not contain the r-subdivided clique of size k as a subgraph.

Generalizes many notions of sparsity such as: bounded degree, bounded treewidth, planarity, excluding a minor, ...

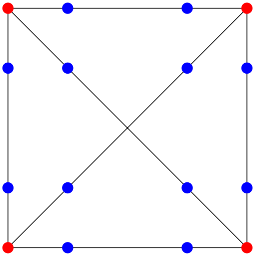


Figure: The 2-subdivided K_4.

Theorem [Grohe, Kreutzer, Siebertz, 2014]

Let $\mathcal{C}$ be a *monotone* class of graphs. If $\mathcal{C}$ is nowhere dense, then FO model checking on $\mathcal{C}$ can be done in time $f(\varphi, \varepsilon) \cdot n^{1+\varepsilon}$ for every $\varepsilon > 0$. Otherwise it is AW[$*$]-hard.

FO Transductions

To go beyond sparse classes, we need to shift from monotone to *hereditary* classes.

FO Transductions

To go beyond sparse classes, we need to shift from monotone to *hereditary* classes.

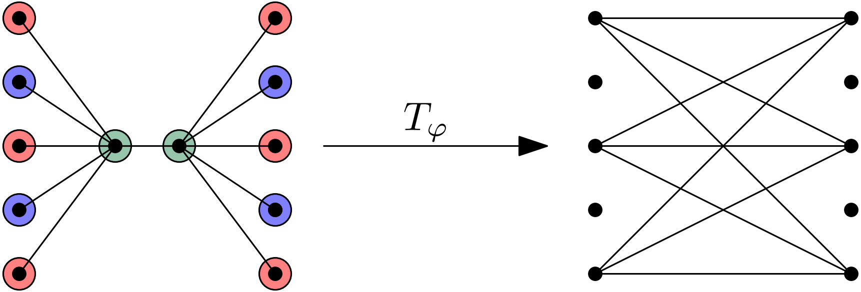
How to produce well behaved hereditary classes from sparse classes?

FO Transductions

To go beyond sparse classes, we need to shift from monotone to *hereditary* classes.

How to produce well behaved hereditary classes from sparse classes?

Transductions $\triangleq$ coloring + interpreting + taking an induced subgraph



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

Structural Sparsity and Monadic Stability

Definition

A class $\mathcal{C}$ is *structurally nowhere dense*, if there exists a transduction T and a nowhere dense class $\mathcal{D}$ such that $\mathcal{C} \subseteq T(\mathcal{D})$.

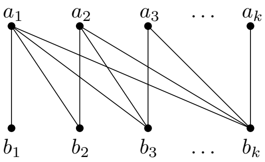
Structural Sparsity and Monadic Stability

Definition

A class $\mathcal{C}$ is *structurally nowhere dense*, if there exists a transduction T and a nowhere dense class $\mathcal{D}$ such that $\mathcal{C} \subseteq T(\mathcal{D})$.

Definition

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



$a_i \sim b_j \Leftrightarrow i \leq j$

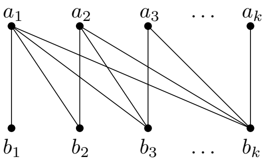
Structural Sparsity and Monadic Stability

Definition

A class $\mathcal{C}$ is *structurally nowhere dense*, if there exists a transduction T and a nowhere dense class $\mathcal{D}$ such that $\mathcal{C} \subseteq T(\mathcal{D})$.

Definition

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

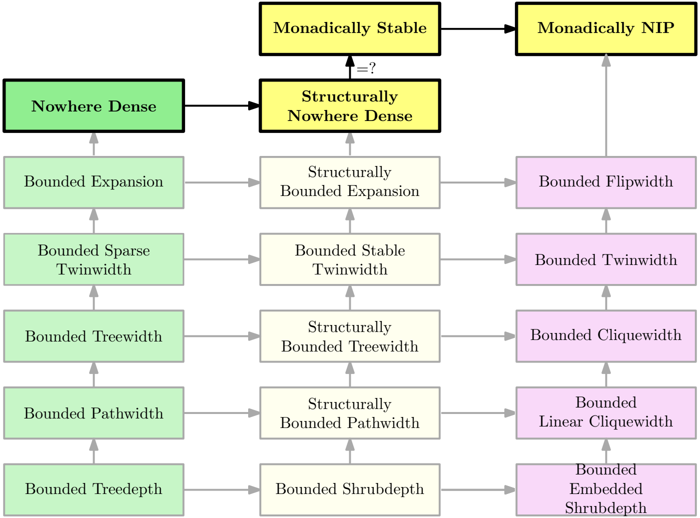


$$a_i \sim b_j \Leftrightarrow i \leq j$$

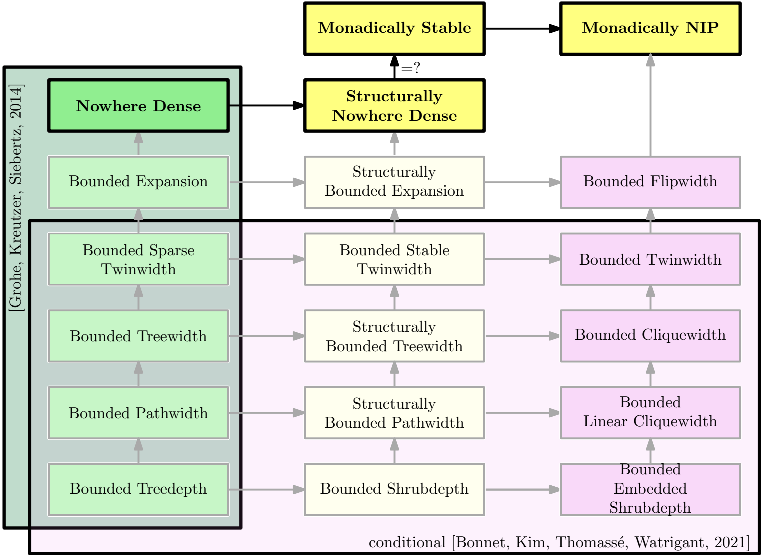
Every structurally nowhere dense class is monadically stable.

Conjecture: every monadically stable class is structurally nowhere dense.

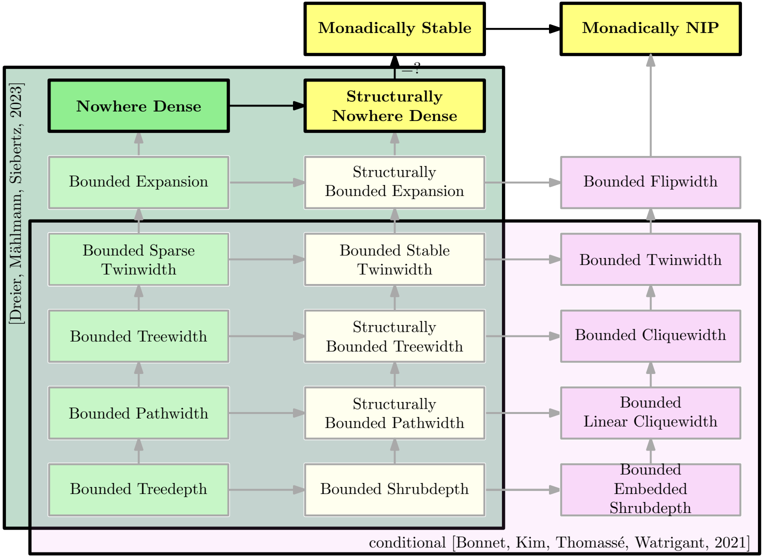
Map of the Universe



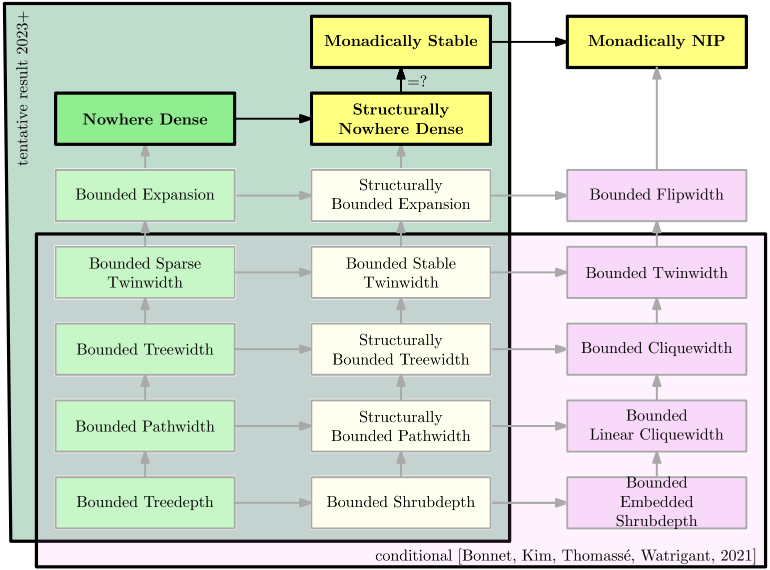
Map of the Universe



Map of the Universe



Map of the Universe



Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

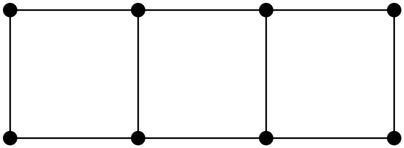
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



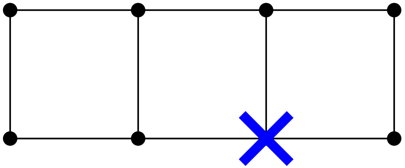
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



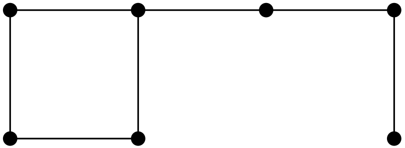
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



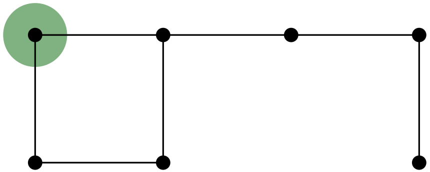
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



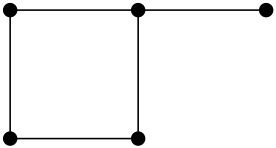
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



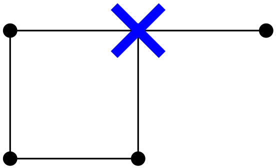
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



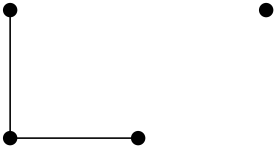
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



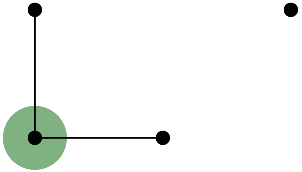
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



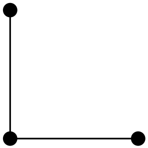
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



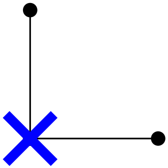
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



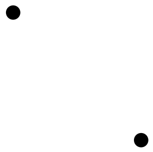
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



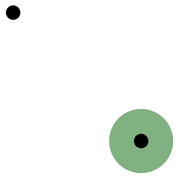
Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:



Splitter Game

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

1. Splitter chooses a vertex v to delete
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i - v$.

Splitter wins once G_i has size 1.

Example play of the radius-2 Splitter game:

The Splitter Game in Nowhere Dense Classes

Theorem [Grohe, Kreutzer, Siebertz, 2013]

A class of graphs $\mathcal{C}$ is nowhere dense $\Leftrightarrow$

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

The Splitter Game in Nowhere Dense Classes

Theorem [Grohe, Kreutzer, Siebertz, 2013]

A class of graphs $\mathcal{C}$ is nowhere dense $\Leftrightarrow$

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

Splitters strategy is efficiently computable and a main ingredient of the nowhere dense model checking.

The Splitter Game in Nowhere Dense Classes

Theorem [Grohe, Kreutzer, Siebertz, 2013]

A class of graphs $\mathcal{C}$ is nowhere dense $\Leftrightarrow$

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

Splitters strategy is efficiently computable and a main ingredient of the nowhere dense model checking.

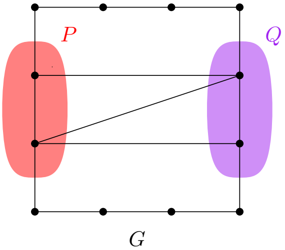
Question: Can we find a similar **game characterization** for monadic stability?

Flips

Denote by $G \oplus (P, Q)$ the graph obtained from G by complementing edges between pairs of vertices from $P \times Q$.

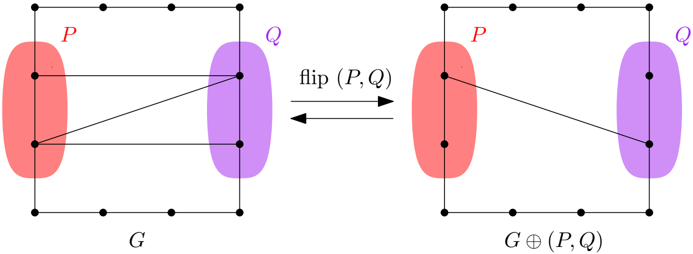
Flips

Denote by $G \oplus (P, Q)$ the graph obtained from G by complementing edges between pairs of vertices from $P \times Q$.



Flips

Denote by $G \oplus (P, Q)$ the graph obtained from G by complementing edges between pairs of vertices from $P \times Q$.



Flipper Game

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

1. Splitter chooses a vertex v
2. Localizer chooses G_{i+1} as a radius-r ball in $G - v$.

Splitter 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 chooses a flip set F
2. Localizer chooses G_{i+1} as a radius-r ball in $G_i \oplus F$.

Flipper wins once G_i has size 1.

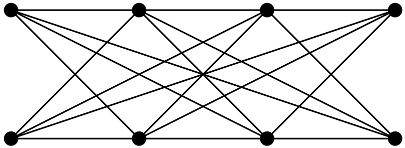
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



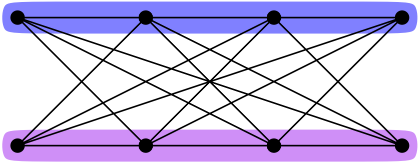
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



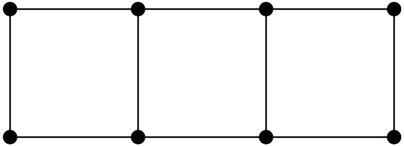
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



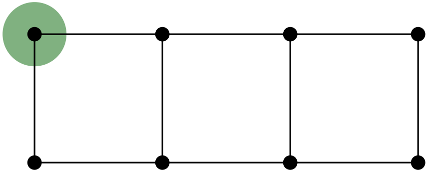
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



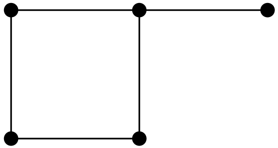
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



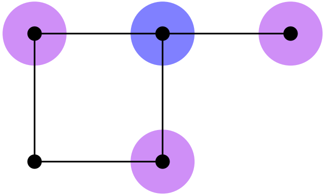
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



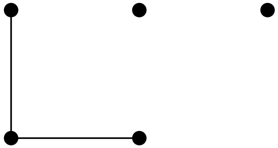
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



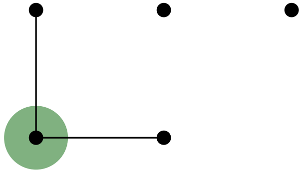
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



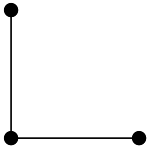
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



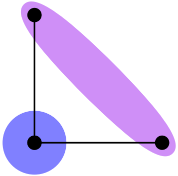
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



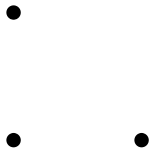
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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:



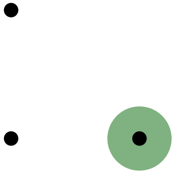
Flipper Game

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

1. Flipper chooses a flip set F
2. Localizer 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 chooses a flip set F
2. Localizer 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:

The Flipper Game in Monadically Stable Classes

Theorem [this paper]

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.

The Flipper Game in Monadically Stable Classes

Theorem [this paper]

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.

Flippers moves are computable in time $\mathcal{O}_{\mathcal{C},r}(n^2)$.

The Flipper Game in Monadically Stable Classes

Theorem [this paper]

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

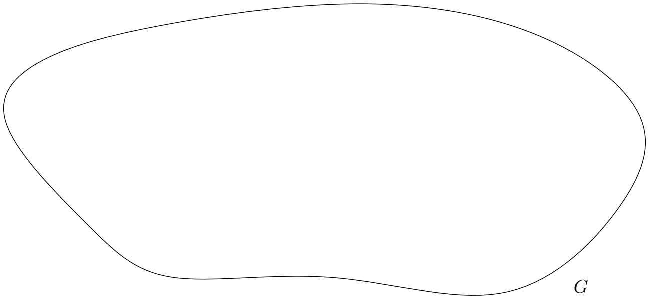
$\forall r \exists \ell$ such that Flipper wins the radius-r game on all graphs from $\mathcal{C}$ in ℓ rounds.

Flippers moves are computable in time $\mathcal{O}_{\mathcal{C},r}(n^2)$.

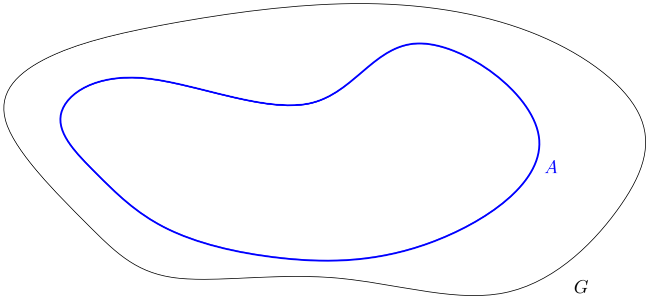
We give two proofs.

1. An algorithmic proof showing: monadic stability $\Rightarrow$ Flipper wins
2. A model theoretic proof showing: monadic stability $\Leftrightarrow$ Flipper wins

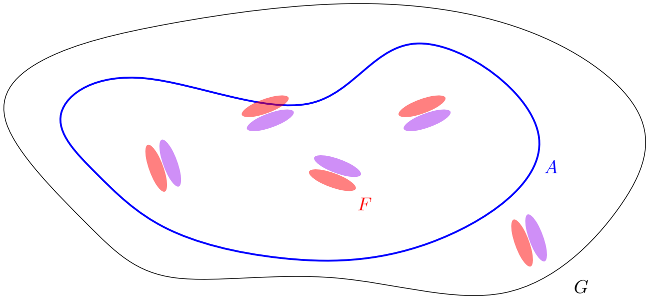
Flip-Flatness



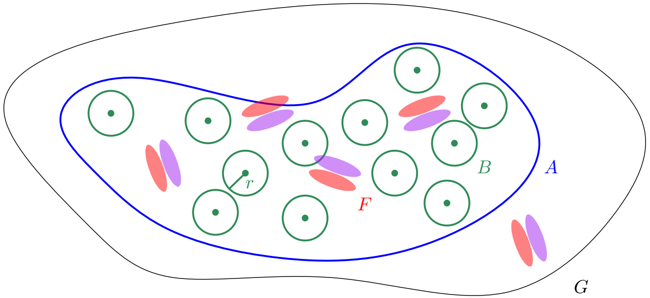
Flip-Flatness



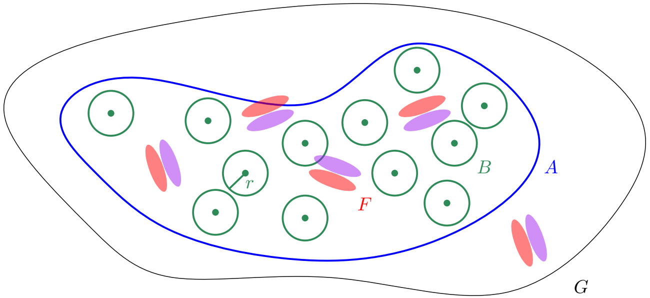
Flip-Flatness



Flip-Flatness



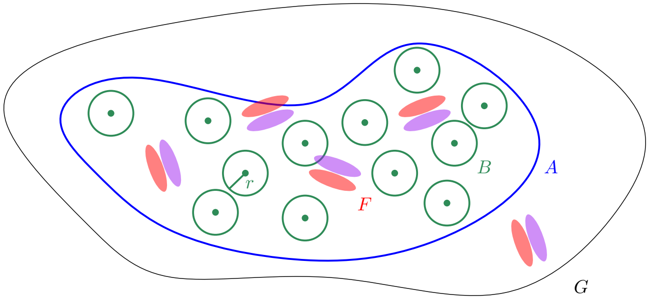
Flip-Flatness



Definition (slightly informal) [Gajarský, Kreutzer]

A class $\mathcal{C}$ is *flip-flat* if for every radius r, in every large set A we find a still large set B that is r-independent after performing a constant number of flips.

Flip-Flatness



Definition (slightly informal) [Gajarský, Kreutzer]

A class $\mathcal{C}$ is *flip-flat* if for every radius r, in every large set A we find a still large set B that is r-independent after performing a constant number of flips.

Theorem [Dreier, Mählmann, Siebertz, Toruńczyk, 2023]

A class $\mathcal{C}$ is flip-flat if and only if it is monadically stable.

Monadic Stability $\Rightarrow$ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability ⇒ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability $\Rightarrow$ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability ⇒ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability ⇒ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability ⇒ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability $\Rightarrow$ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability $\Rightarrow$ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \dots$ be the vertices played by Localizer.

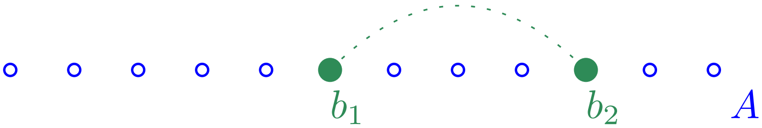
If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

Monadic Stability $\Rightarrow$ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

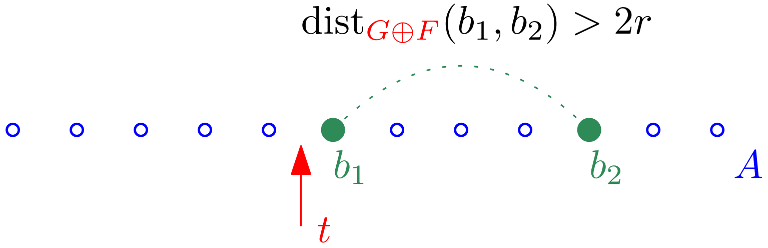
$$\mathrm{dist}_{G \oplus F}(b_1, b_2) > 2r$$



Monadic Stability $\Rightarrow$ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

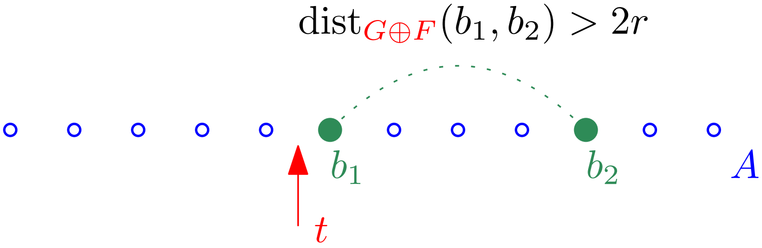
If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.



Monadic Stability ⇒ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.

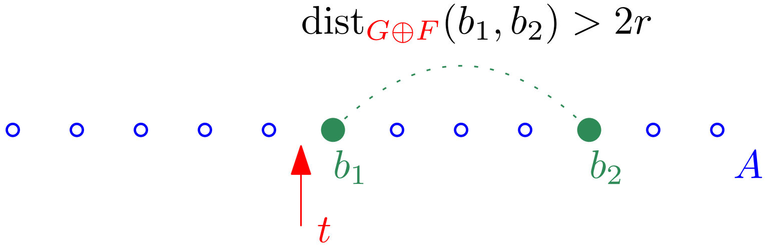


If Flipper had played the flip F at time t then only one of b_1 and b_2 could have survived in the graph.

Monadic Stability ⇒ Flipper Wins - Proof Idea

Let $A = a_1, a_2, a_3, \ldots$ be the vertices played by Localizer.

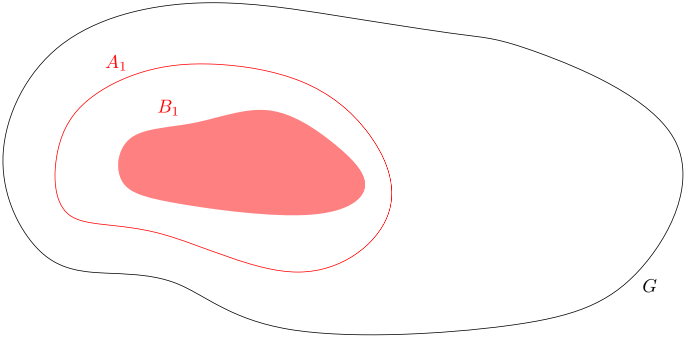
If the game continues long enough, we can apply flip-flatness to find a set $B \subseteq A$ which is $2r$-independent after applying constantly many flips F.



If Flipper had played the flip F at time t then only one of b_1 and b_2 could have survived in the graph.

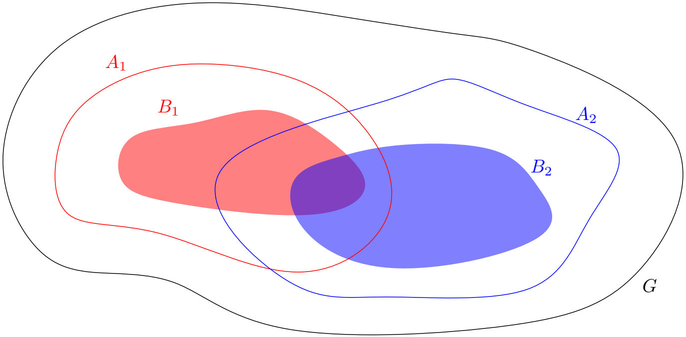
Problem: Flipper does not know A at time t.

Predictable Flip-Flatness



$\text{ff}(A_1) = (B_1, F_1)$

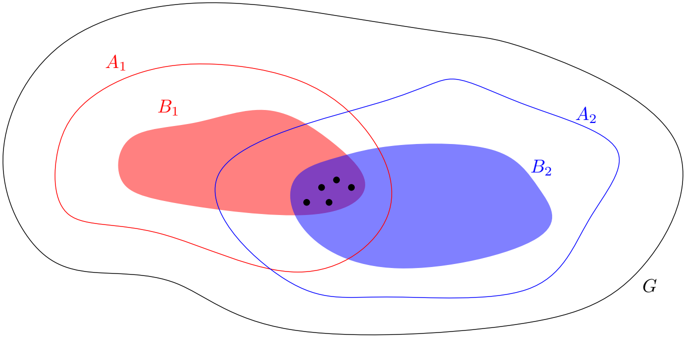
Predictable Flip-Flatness



$\mathrm{ff}(A_1) = (B_1, F_1)$

$\mathrm{ff}(A_2) = (B_2, F_2)$

Predictable Flip-Flatness



$\mathrm{ff}(A_1) = (B_1, F_1)$

$\mathrm{ff}(A_2) = (B_2, F_2)$

$|B_1 \cap B_2| \geq 5 \quad \Rightarrow \quad F_1 = F_2$

$F_1 = F_2$ are computable from a five-element subset of $B_1 \cap B_2$ in time $\mathcal{O}(n^2)$.

Flippers Winning Strategy

For every 5 element subset P of Localizers previous moves:

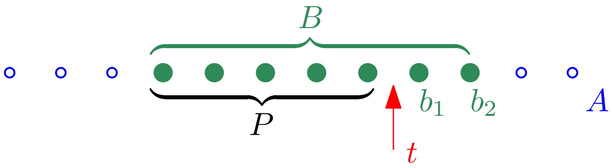
1. apply the flips $\mathrm{predict}(P)$ for radius $2r$
2. let Localizer localize to an r-ball
3. undo $\mathrm{predict}(P)$

Flippers Winning Strategy

For every 5 element subset P of Localizers previous moves:

1. apply the flips $\mathrm{predict}(P)$ for radius $2r$
2. let Localizer localize to an r-ball
3. undo $\mathrm{predict}(P)$

Assume Localizer can play enough rounds to apply size 7 flip-flatness

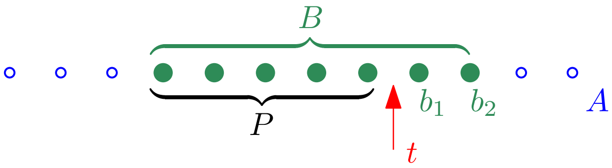


Flippers Winning Strategy

For every 5 element subset P of Localizers previous moves:

1. apply the flips $\mathrm{predict}(P)$ for radius $2r$
2. let Localizer localize to an r-ball
3. undo $\mathrm{predict}(P)$

Assume Localizer can play enough rounds to apply size 7 flip-flatness



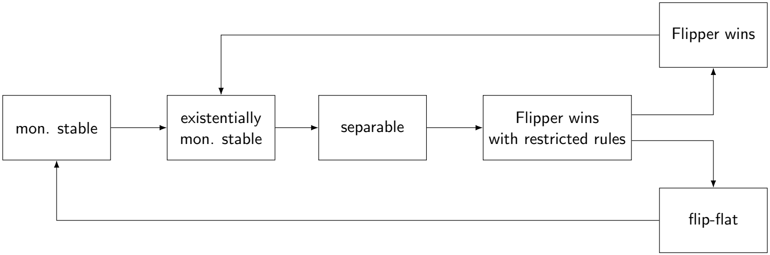
At time t, P was considered as a subset of Localizers previous moves.

B was flipped $2r$-independent and only one of b_1, b_2 survived. Contradiction!

Monadic Stability ⇔ Flipper Wins

How to prove Flipper wins ⇒ monadic stability?

The model theoretic proof unravels further characterizations!



Existential Monadic Stability

Definition

A class is *existentially monadically stable*, if it does not transduce the class of all half graphs using an **existential formula**.

A formula is existential if it can be written as

$$\exists x_1, \ldots, x_k \psi(x_1, \ldots, x_k)$$

where ψ is quantifier free.

Existential Monadic Stability

Definition

A class is *existentially monadically stable*, if it does not transduce the class of all half graphs using an **existential formula**.

A formula is existential if it can be written as

$$\exists x_1, \ldots, x_k \psi(x_1, \ldots, x_k)$$

where ψ is quantifier free.

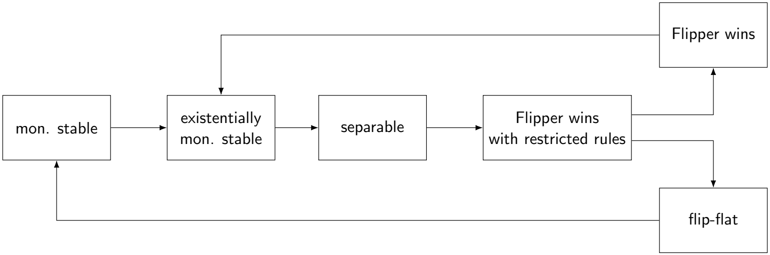
This is a "weaker" condition than monadic stability, so it is "easier" to show

Flipper wins $\Rightarrow$ existential monadic stability

Monadic Stability ⇔ Flipper Wins

How to prove Flipper wins ⇒ monadic stability?

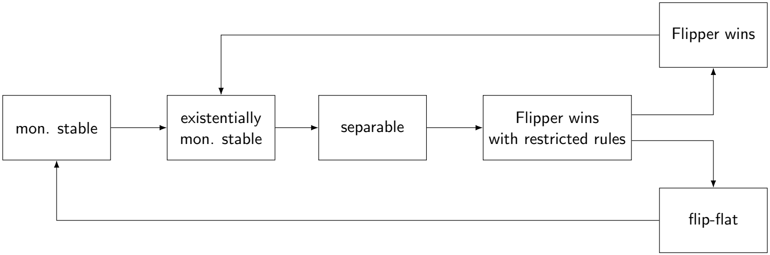
The model theoretic proof unravels further characterizations!



Monadic Stability ⇔ Flipper Wins

How to prove Flipper wins ⇒ monadic stability?

The model theoretic proof unravels further characterizations!



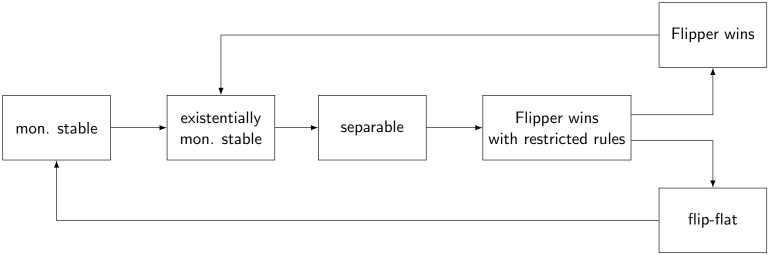
Separability is a model theoretic property.

We show separability ⇒ Flipper wins by a compactness argument.

Monadic Stability ⇔ Flipper Wins

How to prove Flipper wins ⇒ monadic stability?

The model theoretic proof unravels further characterizations!



Separability is a model theoretic property.

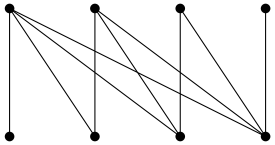
We show separability ⇒ Flipper wins by a compactness argument.

We use a stricter game variant which allows us to recover flip-flatness.

Summary

Definition

A class is *monadically stable* if it does not transduce the class of all half graphs using FO logic.



The Flipper game

- formalizes the process of recursive decomposition by flips and localizations,
- characterizes monadic stability,
- is analogous to a game characterization of nowhere density,
- can be proven using methods from either combinatorics or model theory,
- is a key ingredient for algorithmic applications, e.g. FO model checking.