Flipping and Forking

LICS 2025

Wojciech Przybyszewski and Szymon Toruńczyk

University of Warsaw

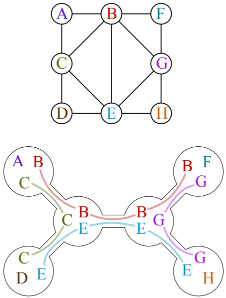
26th June 2025

Tree-like decompositions

Everybody loves tree-like decompositions of graphs!

Tree-like decompositions

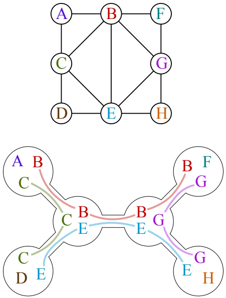
Everybody loves tree-like decompositions of graphs!



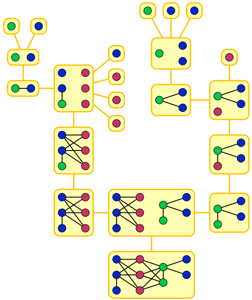
Treewidth

Tree-like decompositions

Everybody loves tree-like decompositions of graphs!



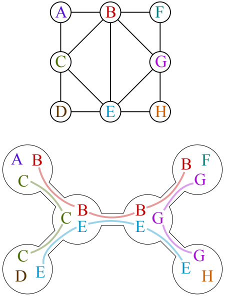
Treewidth



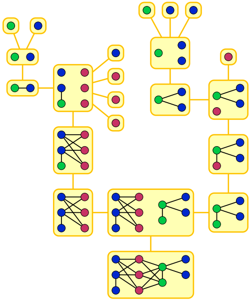
Clique-width

Tree-like decompositions

Everybody loves tree-like decompositions of graphs!



Treewidth



Clique-width

- Useful for dynamic programming on graphs
- Enable efficient model checking algorithms
- Often yield fixed-parameter tractable (FPT) algorithms

Figures adapted from Wikipedia (David Eppstein)

Monadically stable classes of graphs

Definition

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

Monadically stable classes of graphs

Definition

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

Definition

A half-graph of order n is a bipartite graph (U, V, E) with $U = \{a_1, \ldots, a_n\}, V\{b_1, \ldots, b_n\}$ and edges $E = \{(a_i, b_j) \mid i \leqslant j\}$.

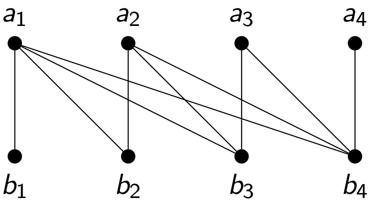


Figure: A half-graph of order $n = 4$.

Monadically stable classes of graphs

Definition

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

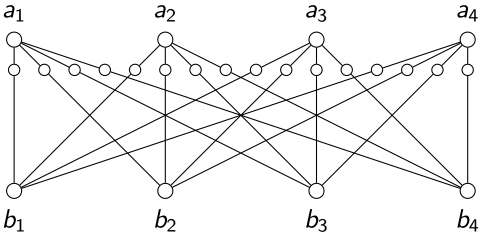


Figure: 1-subdivision of $K_{4,4}$.

Monadically stable classes of graphs

Definition

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

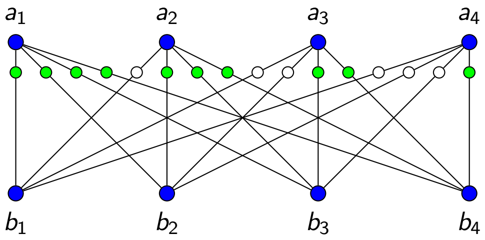
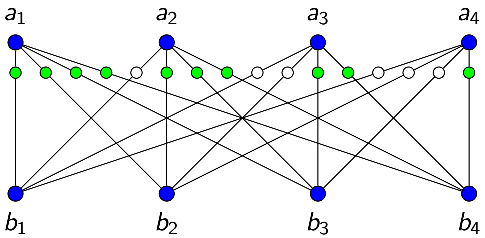


Figure: Colored 1-subdivision of $K_{4,4}$.

Monadically stable classes of graphs

Definition

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



Now let's apply

$$\varphi(x, y) \equiv \mathrm{Blue}(x) \wedge \mathrm{Blue}(y) \wedge \exists z \, (\mathrm{Green}(z) \wedge E(x, z) \wedge E(y, z)) \, .$$

Monadically stable classes of graphs

Definition

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

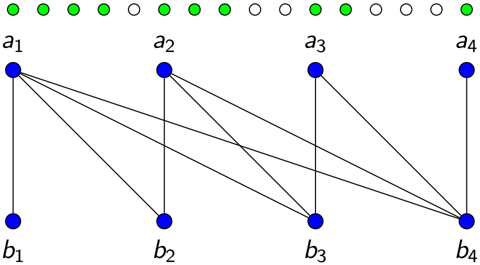


Figure: The result of applying φ to the colored graph.

Monadically stable classes of graphs

Definition

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

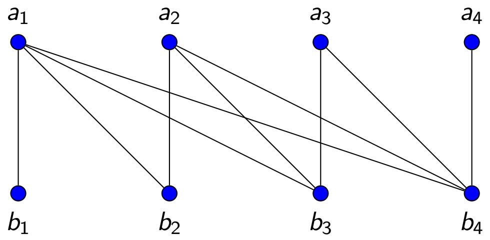
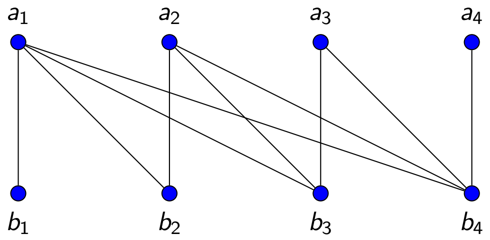


Figure: We take an induced subgraph.

Monadically stable classes of graphs

Definition

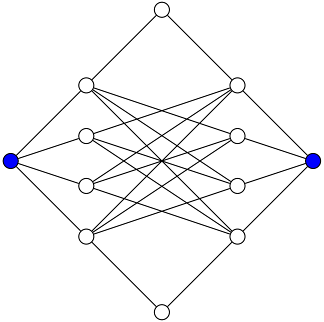
A class of graphs is *monadically stable* if it does not transduce the class of all half-graphs.



Fact

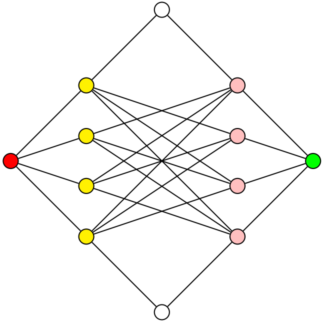
The class of all 1-subdivided bicliques is not monadically stable.

(Definable) flips



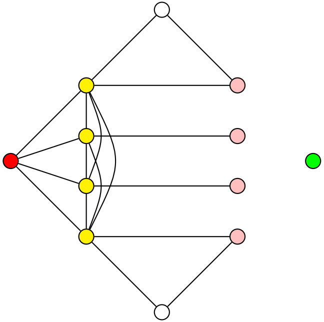
We start with a graph and a subset S of its vertices (in blue).

(Definable) flips



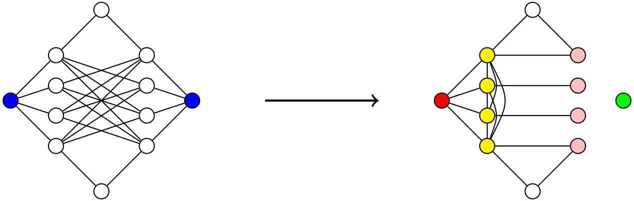
We partition the vertices of $V(G)$ by their neighborhoods on S.
Let's flip the edges between these pairs of colors: (○, ○), (○, ○), (○, ○).

(Definable) flips



The result of the flip.

(Definable) flips



Definition

Fix a graph G and a set $S \subseteq V(G)$. An S-flip of G is obtained as follows:
1. Partition $V(G)$ by their neighborhoods in S;
2. For each pair or parts (P_1, P_2) (possibly $P_1 = P_2$) either keep the edges between P_1 and P_2 or complement them.

Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

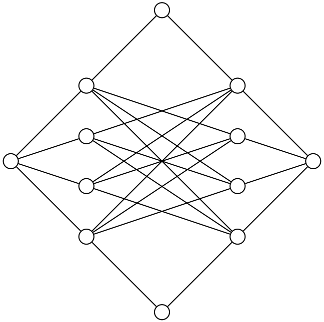
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



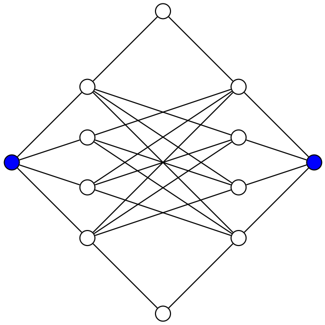
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



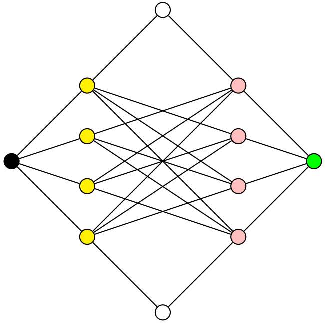
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



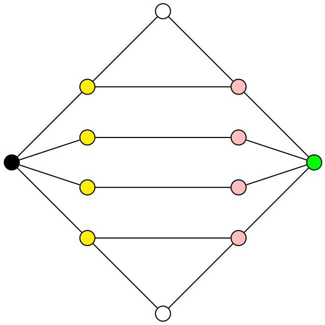
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



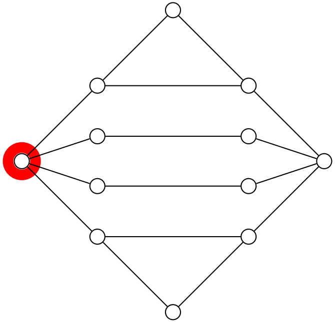
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



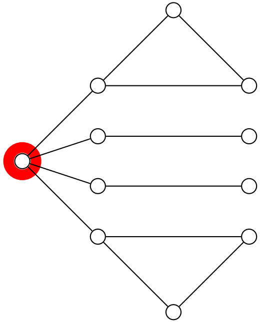
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



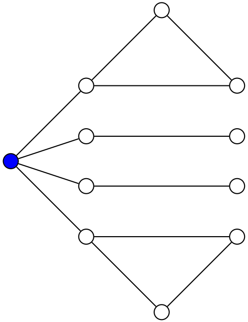
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



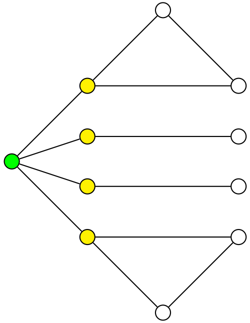
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



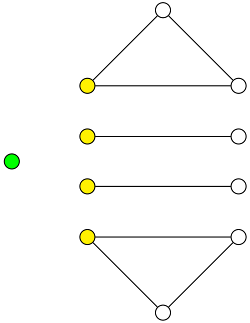
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



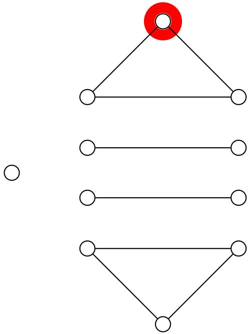
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



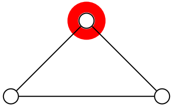
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



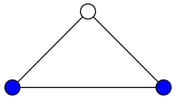
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



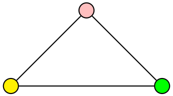
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



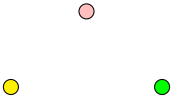
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



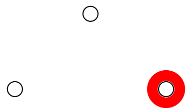
Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:



Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Example play of the radius-2 batch-2 Flipper game:

Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Theorem [Gajarský, Mählmann, McCarty, Ohlmann, Pilipczuk, P., Siebertz, Sokołowski, Toruńczyk, '23]

The following are equivalent for a class of graphs $\mathcal{C}$:

- $\mathcal{C}$ is monadically stable,
- $\forall r\ \exists k\ \exists \ell$ such that for every graph $G \in \mathcal{C}$ Flipper wins the radius-r batch-ℓ Flipper game on G in at most k rounds.

Flipper Game

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

1. Flipper chooses a set S_i with $|S_i| \leqslant \ell$ and an S_i-flip H_i of G_i.
2. Connector chooses G_{i+1} as an induced subgraph given by radius-r ball in H_i.

Flipper wins once G_i has size 1.

Theorem [Gajarský, Mählmann, McCarty, Ohlmann, Pilipczuk, P., Siebertz, Sokołowski, Toruńczyk, '23]

The following are equivalent for a class of graphs $\mathcal{C}$:

- $\mathcal{C}$ is monadically stable,
- $\forall r\ \exists k\ \exists \ell$ such that for every graph $G \in \mathcal{C}$ Flipper wins the radius-r batch-ℓ Flipper game on G in at most k rounds.

This gives tree-like decompositions for monadically stable classes of graphs.

Monadically stable classes of relational structures

Definition

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

Monadically stable classes of relational structures

Definition

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

Monadically stable classes of relational structures

Definition

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

Our contribution:

- We give a definition of (definable) flips for relational structures.

Monadically stable classes of relational structures

Definition

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

Our contribution:

- We give a definition of (definable) flips for relational structures.
- We characterize monadically stable classes of relational structures combinatorially, in terms of flipper game.

Monadically stable classes of relational structures

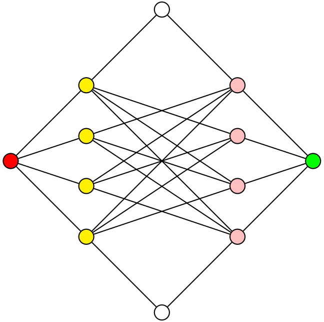
Definition

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

Our contribution:

- We give a definition of (definable) flips for relational structures.
- We characterize monadically stable classes of relational structures combinatorially, in terms of flipper game.
- We characterize forking independence – a fundamental notion in model theory – in terms of flips.

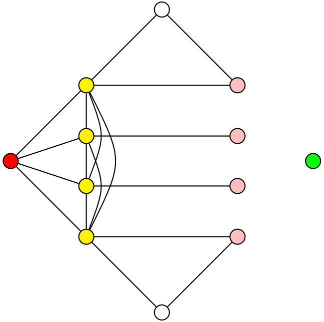
What are flips?



Observation

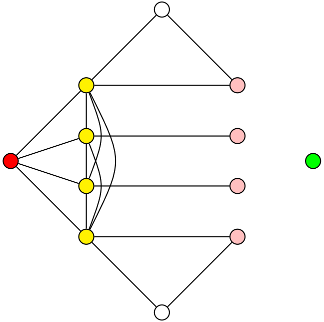
The edges of an S-flip can be defined by a quantifier free formula $\varphi(x, y)$ with parameters from S.

What are flips?



Flips are reversible – the edges of the original graph can be defined in a flip by a quantifier free formula $\psi(x, y)$ with parameters from S.

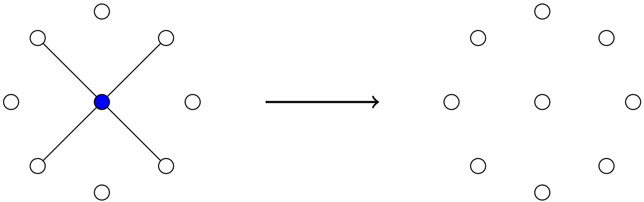
What are flips?



Flips are reversible – the edges of the original graph can be defined in a flip by a quantifier free formula $\psi(x, y)$ with parameters from S.

Are they?

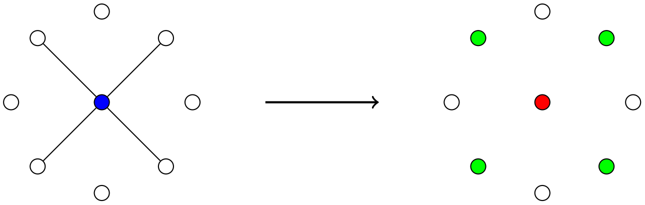
What are flips?



Flips are reversible – the edges of the original graph can be defined in a flip by a quantifier free formula $\psi(x, y)$ with parameters from S.

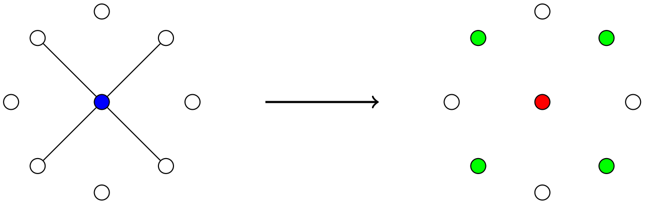
Are they?

What are flips?



Flips are reversible – the edges of the original graph can be defined in a flip by a quantifier free formula $\psi(x, y)$ with parameters from S.
Are they?

What are flips?



Flips are reversible – the edges of the original graph can be defined in a flip by a quantifier free formula $\psi(x, y)$ with parameters from S.
We need the access to the neighborhood classes in the original graph. Luckily, they are also definable by a quantifier free formula!

Flips of relational structures

Definition

Fix a Γ-structure $\mathbf{N}$ and a Σ-structure $\mathbf{M}$ on the same universe. We say that $\mathbf{N}$ is a flip of $\mathbf{M}$ if they are quantifier-free bi-interpretable with parameters:

Flips of relational structures

Definition

Fix a Γ-structure **N** and a Σ-structure **M** on the same universe. We say that **N** is a flip of **M** if they are quantifier-free bi-interpretable with parameters:

For each $R(\bar{x}) \in \Gamma$ there exists $\varphi_R(\bar{x})$ a quantifier-free Σ-formula with parameters s.t.

$$\mathbf{N} \models R(\bar{a}) \iff \mathbf{M} \models \varphi_R(\bar{a}) \quad \text{for every } \bar{a} \in M^{\bar{x}}$$

Flips of relational structures

Definition

Fix a Γ-structure $\mathbf{N}$ and a Σ-structure $\mathbf{M}$ on the same universe. We say that $\mathbf{N}$ is a flip of $\mathbf{M}$ if they are quantifier-free bi-interpretable with parameters:

For each $R(\bar{x}) \in \Gamma$ there exists $\varphi_R(\bar{x})$ a quantifier-free Σ-formula with parameters s.t.

$$\mathbf{N} \models R(\bar{a}) \iff \mathbf{M} \models \varphi_R(\bar{a}) \quad \text{for every } \bar{a} \in M^{\bar{x}}$$

and for each $T(\bar{y}) \in \Sigma$ there exists $\psi_T(\bar{y})$ a q.f. Γ-formula with parameters s.t.

$$\mathbf{M} \models T(\bar{b}) \iff \mathbf{N} \models \psi_T(\bar{b}) \quad \text{for every } \bar{b} \in M^{\bar{y}}.$$

Flips of relational structures

Definition

Fix a Γ-structure $\mathbf{N}$ and a Σ-structure $\mathbf{M}$ on the same universe. We say that $\mathbf{N}$ is a flip of $\mathbf{M}$ if they are quantifier-free bi-interpretable with parameters:

For each $R(\bar{x}) \in \Gamma$ there exists $\varphi_R(\bar{x})$ a quantifier-free Σ-formula with parameters s.t.

$$\mathbf{N} \models R(\bar{a}) \iff \mathbf{M} \models \varphi_R(\bar{a}) \quad \text{for every } \bar{a} \in M^{\bar{x}}$$

and for each $T(\bar{y}) \in \Sigma$ there exists $\psi_T(\bar{y})$ a q.f. Γ-formula with parameters s.t.

$$\mathbf{M} \models T(\bar{b}) \iff \mathbf{N} \models \psi_T(\bar{b}) \quad \text{for every } \bar{b} \in M^{\bar{y}}.$$

We say that $\mathbf{N}$ is an S-flip of $\mathbf{M}$ if all the formulas mentioned above use only parameters from S.

Example

Fix a rooted tree T. Take $\mathbf{M} = (V(T), \{(u, v, w) \mid u = \mathrm{lca}(v, w)\})$.

Example

Fix a rooted tree T. Take $\mathbf{M} = (V(T), \{(u, v, w) \mid u = \mathrm{lca}(v, w)\})$.
Goal: Isolate a given vertex s.

Example

Fix a rooted tree T. Take $\mathbf{M} = (V(T), \{(u, v, w) \mid u = \mathrm{lca}(v, w)\})$.
Goal: Isolate a given vertex s.
Define $\mathbf{N} = (V(T), R_0, R_1, R_2)$ where

$$R_0 = \{(u, v, w) \in R \mid u, v, w \neq s\},$$

$$R_1 = \{(w, v) \mid (s, w, v) \in R\},$$

$$R_2 = \{(u, w) \mid (u, s, w) \in R\}.$$

Example

Fix a rooted tree T. Take $\mathbf{M} = (V(T), \{(u, v, w) \mid u = \mathrm{lca}(v, w)\})$.
Goal: Isolate a given vertex s.
Define $\mathbf{N} = (V(T), R_0, R_1, R_2)$ where

$$R_0 = \{(u, v, w) \in R \mid u, v, w \neq s\},$$

$$R_1 = \{(w, v) \mid (s, w, v) \in R\},$$
$$R_2 = \{(u, w) \mid (u, s, w) \in R\}.$$

Claim

$\mathbf{N}$ is an $\{s\}$-flip of $\mathbf{M}$.

Example

Fix a rooted tree T. Take $\mathbf{M} = (V(T), \{(u, v, w) \mid u = \mathrm{lca}(v, w)\})$.
Goal: Isolate a given vertex s.
Define $\mathbf{N} = (V(T), R_0, R_1, R_2)$ where

$$R_0 = \{(u, v, w) \in R \mid u, v, w \neq s\},$$

$$R_1 = \{(w, v) \mid (s, w, v) \in R\},$$
$$R_2 = \{(u, w) \mid (u, s, w) \in R\}.$$

Claim

$\mathbf{N}$ is an $\{s\}$-flip of $\mathbf{M}$.

$$\mathbf{M} \models R(x, y, z) \iff \mathbf{N} \models R_0(x, y, z) \vee (x = s \wedge R_1(y, z)) \vee (y = s \wedge R_2(x, z)) \vee (z = s \wedge R_2(x, y)).$$

Flipper game for relational structures

Theorem [P., Toruńczyk]

The following are equivalent for a class of relational structures $\mathcal{C}$:

- $\mathcal{C}$ is monadically stable,
- $\forall r \; \exists k$ such that for every structure $G \in \mathcal{C}$ Flipper wins the radius-r Flipper game on G in at most k rounds.

Flipping and forking

Forking independence is an abstract notion of independence for arbitrary structures introduced by Shelah.

It generalizes e.g. linear independence in vector spaces and algebraic independence in fields.

Flipping and forking

Forking independence is an abstract notion of independence for arbitrary structures introduced by Shelah.

It generalizes e.g. linear independence in vector spaces and algebraic independence in fields.

For $a, b \in \mathbf{M}, C \subseteq \mathbf{M}$

$$a \mathop{\smile\!\!\!\!\!\!|}_{C} b$$

means that the element a is forking independent from b over the set C.

Flipping and forking

For $a, b \in \mathbf{M}$, $C \subseteq \mathbf{M}$

$$a \underset{C}{\perp} b$$

means that the element a is forking independent from b over the set C.

Definition [GMMOPPSST'23, PT'25]

We say that *a is distance-r flip independent from b over C* (denoted $a \perp^r_C b$) if there is a C-flip $\mathbf{N}$ of $\mathbf{M}$ such that the distance between a and b in the Gaifman graph of $\mathbf{N}$ is $\geqslant r$.

Flipping and forking

For $a, b \in \mathbf{M}$, $C \subseteq \mathbf{M}$

$$a \underset{C}{\perp\!\!\!\smile} b$$

means that the element a is forking independent from b over the set C.

Definition [GMMOPPSST'23, PT'25]

We say that *a is distance-r flip independent from b over C* (denoted $a \perp\!\!\!\smile^{r}_{C} b$) if there is a C-flip $\mathbf{N}$ of $\mathbf{M}$ such that the distance between a and b in the Gaifman graph of $\mathbf{N}$ is $\geqslant r$.

Theorem [P., Toruńczyk]

For every monadically stable structure $\mathbf{M}$, its elementary extension $\mathbf{N}$, and two distinct elements $a, b \in \mathbf{N}$

$$a \underset{\mathbf{M}}{\perp\!\!\!\smile} b \iff \forall r \quad a \overset{r}{\underset{\mathbf{M}}{\perp\!\!\!\smile}} b.$$

Summary

- We defined flips for relational structures as **quantifier-free bi-interpretations with parameters**.

Summary

- We defined flips for relational structures as **quantifier-free bi-interpretations with parameters**.
- We gave a combinatorial characterization of monadically stable classes of relational structures which yields tree-like decompositions.

Summary

- We defined flips for relational structures as **quantifier-free bi-interpretations with parameters**.
- We gave a combinatorial characterization of monadically stable classes of relational structures which yields tree-like decompositions.
- We characterized forking independence in terms of flips.

Summary

- We defined flips for relational structures as **quantifier-free bi-interpretations with parameters**.
- We gave a combinatorial characterization of monadically stable classes of relational structures which yields tree-like decompositions.
- We characterized forking independence in terms of flips.

Thank you!