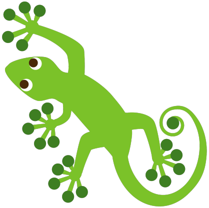
First Order Model Checking in sparse classes

Hugo Jacob

AlGCo, LIRMM, Université de Montpellier



Vocabulary

- Vocabulary Σ: The set of symbols used for the relations and their arity
- Structure $\mathbb{A}$: A vocabulary, a vertex set, and a relation for each symbol of the vocabulary
- Formula: A first order formula over a vocabulary is inductively defined as either an atomic formula: $x = y$ or $R(x_1, \ldots, x_k)$; or a combination of formulae by boolean connectives and quantifiers:
  $\neg\varphi \qquad \varphi \vee \psi \qquad \varphi \wedge \psi \qquad \forall x.\varphi(x) \qquad \exists x.\psi(x)$
- Variables: bound/free
- Gaifman graph: edge between vertices with a common tuple of a relation

Basic algebra

$$\varphi \wedge \psi \quad \rightarrow \quad \neg(\neg\varphi \vee \neg\psi)$$

$$\forall x.\varphi(x) \quad \rightarrow \quad \neg\exists x.\neg\varphi(x)$$

$$(A \vee B) \wedge C \quad \rightarrow \quad (A \wedge C) \vee (B \wedge C)$$

First Order Model Checking

The goal:

Theorem

For every Σ-structure $\mathbb{A}$ in $\mathcal{C}$, for any FO$[\Sigma]$ sentence φ, we can decide $\mathbb{A} \models \varphi$ in time $f(\mathcal{C}, \varphi) \cdot |\mathbb{A}|^{O(1)}$.

Roadmap

Algorithm for:

- Forests of bounded depth
- Structures of bounded treedepth
- Structures of bounded expansion

Roadmap

Algorithm for:

- Forests of bounded depth
- Structures of bounded treedepth
- Structures of bounded expansion

Further results:

- Nowhere dense structures
- Extensions to the dense setting

Forests of bounded depth

Forests: We consider relations of arity at most one (decorations on vertices) and the `parent` function.

Plan:

- Induction on formula
- Consider formulae in normal form
- Enrich the forest with decorations.
- Dynamic programming to compute decorations.

Induction statement

Lemma

In the class of forests of depth d, for any forest $\mathbb{F}$ and any FO formula φ, in time $f(d,\varphi) \cdot ||\mathbb{F}||$, we can compute a **quantifier-free** *formula $\widehat{\varphi}$ together with a 'decorated' forest $\mathbb{F}_{\widehat{\varphi}}$ such that:*

$$\mathbb{F}, \overline{a} \models \varphi(\overline{a}) \Leftrightarrow \mathbb{F}_{\widehat{\varphi}}, \overline{a} \models \widehat{\varphi}(\overline{a})$$

Algorithm

Simple cases

- $\neg$: Set $\widehat{(\varphi)} := \neg\widehat{\psi}$ and keep the decorated forest used for ψ
- $\vee$: Set $\widehat{\varphi} := \widehat{\psi_1} \vee \widehat{\psi_2}$ and $\mathbb{F}_{\widehat{\varphi}} = \mathbb{F}_{\widehat{\psi_1}} \cup \mathbb{F}_{\widehat{\psi_2}}$.
- atomic formulae: Already quantifier free, we keep φ and $\mathbb{F}$.

Quantification

We consider the case $\varphi(\overline{x}) = \exists y.\psi(\overline{x}, y)$.
Induction hypothesis: $\widehat{\psi}$ and $\mathbb{F}_{\widehat{\psi}}$.
$\widehat{\psi}$ is **quantifier-free**, we can consider an equivalent ψ':

$$\psi' = \bigvee_{\mathbb{P} \vDash \widehat{\psi}} \psi_{\mathbb{P}}$$

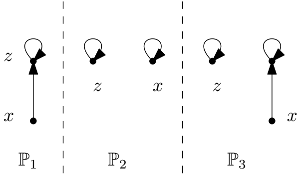
A pattern $\mathbb{P}$ is a small structure induced by the ancestors of the variables.
We enumerate all possible structures that can satisfy $\widehat{\psi}$

Algorithm

Quantification

We consider the case $\varphi(\overline{x}) = \exists y.\psi(\overline{x}, y)$.
Induction hypothesis: $\widehat{\psi}$ and $\mathbb{F}_{\widehat{\psi}}$.
$\widehat{\psi}$ is **quantifier-free**, we can consider an equivalent ψ':

$$\psi' = \bigvee_{\mathbb{P} \models \widehat{\psi}} \psi_{\mathbb{P}}$$

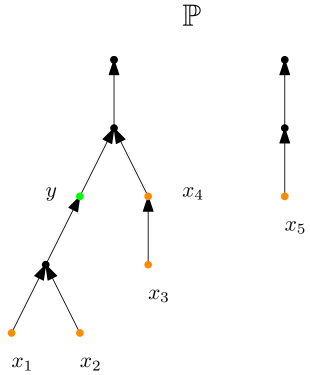


$\psi' = (\psi_{\mathbb{P}_1} \vee \psi_{\mathbb{P}_2} \vee \psi_{\mathbb{P}_3})$

Patterns of depth 2 satisfying:
$\widehat{\psi} = (z = \texttt{parent}(z) \wedge \neg(z = x))$

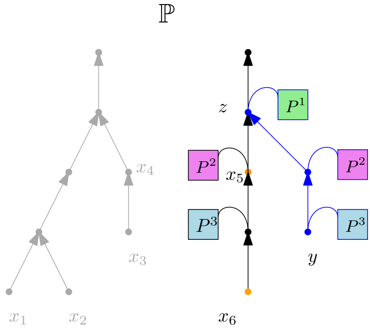
Quantification continued

If y is an ancestor of a vertex x of $\overline{x}$ in $\mathbb{P}$, then replace y by $\texttt{parent}^i(x)$ in $\widehat{\psi}$ to obtain $\widehat{\varphi}$.



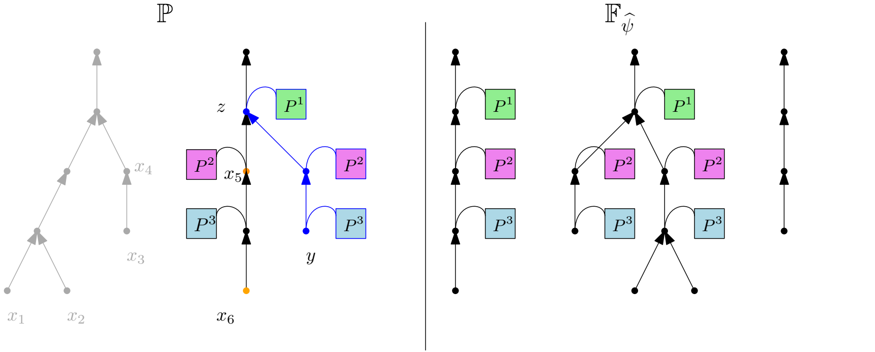
Quantification continued

If y has a common ancestor z with a vertex of $\overline{x}$ in $\mathbb{P}$,



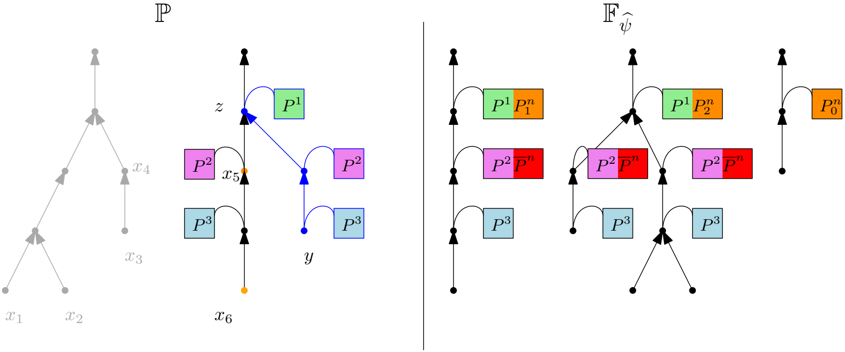
Quantification continued

If y has a common ancestor z with a vertex of $\overline{x}$ in $\mathbb{P}$,



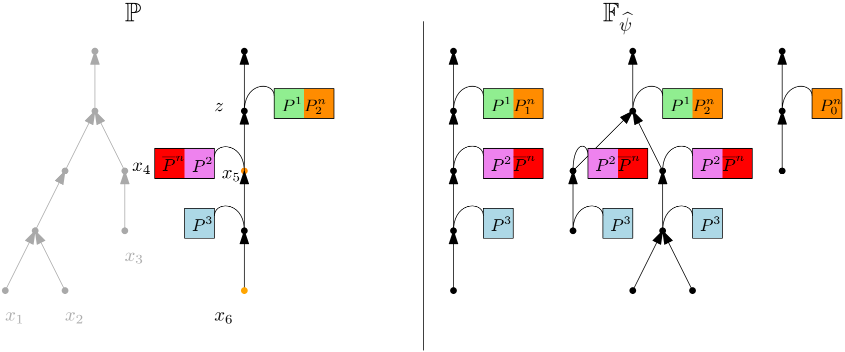
Quantification continued

If y has a common ancestor z with a vertex of $\overline{x}$ in $\mathbb{P}$,



Algorithm

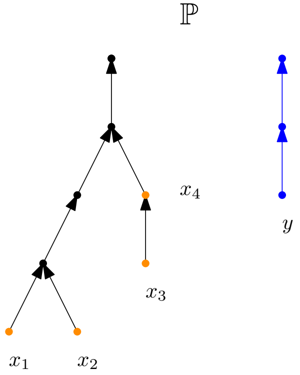
Quantification continued



Algorithm

Quantification continued

If y has no common ancestor with vertices of $\overline{x}$ in $\mathbb{P}$,



Dynamic programming on (rooted) forests

To compute the new predicates, we can use simple dynamic programming on the decorated forest $\widehat{\mathbb{F}}$. We are always looking for a fixed decorated path $(y = p_0, p_1, \ldots, p_\ell = z)$ (from $\mathbb{P}$). Let n be the depth of y and m be the depth of z.

We compute the boolean function $C : V \to \{\bot, \top\}$ defined by:

- At depth n: $C(v) = \top$ iff v satisfies exactly the same predicates as y in $\mathbb{P}$.
- At depth $i \in [m, n[$: $C(v) = \top$ iff there is a child w of v such that $C(w) = \top$ and v satisfies exactly the same predicates as p_{i-n}

This is easily computed in time $f(\varphi, d) \cdot |V|$

Running time

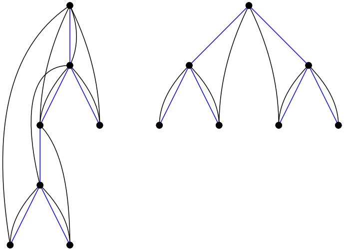
For each case of the induction, we use at most $f(\varphi, d) \cdot |V|$ time.

We conclude that we can perform model-checking on forests of depth d in time $f(\varphi, d) \cdot ||\mathbb{F}||$.

Structures of bounded treedepth

Definition

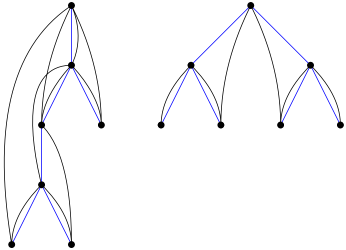
The treedepth of a graph is the minimum depth of a forest whose ancestor-descendent relation covers all edges.



Structures of bounded treedepth

Definition

The treedepth of a graph is the minimum depth of a forest whose ancestor-descendent relation covers all edges.



We now consider a structure $\mathbb{A}$ of bounded treedepth: its Gaifman graph has bounded treedepth.

Obtaining a forest

Any Depth First Search tree of the Gaifman graph is a valid decomposition.

Obtaining a forest

Any Depth First Search tree of the Gaifman graph is a valid decomposition.

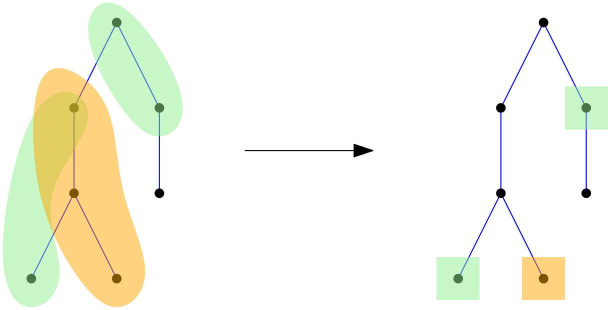
A graph of treedepth k contains no path on 2^k vertices

Let d be the depth of our computed forest.

Translating the formula

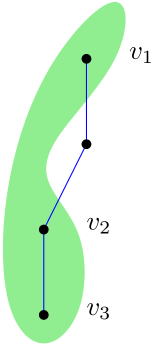
We now reduce to the setting of the previous algorithm.

All we need to do is to translate the relations of any arity as unary relations on the forest.



Translating the formula

We fix a relation R, and a tuple $(v_1, \dots, v_k) \in R$ and we want to replace this information with unary relations.



$$(v_1, v_2, v_3) \in R$$

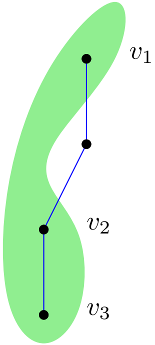
$$\delta(1) = 3, \delta(2) = 1, \delta(3) = 0$$

Translating the formula

distance vector $\delta : [1,k] \to [0,d]$ s.t.
$\exists i \in [1,k].\ \delta(i) = 0$

$$v \in R^\delta := (\texttt{parent}^{\delta(1)}(v), \ldots, \texttt{parent}^{\delta(k)}(v)) \in R$$

$$R(\overline{x}) \to \bigvee_\delta \left(R^\delta(v) \wedge \bigwedge_{j=1}^{k} \texttt{parent}^{\delta(j)}(v) = x_j \right)$$



$(v_1, v_2, v_3) \in R$
$\delta(1) = 3, \delta(2) = 1, \delta(3) = 0$

Running time

We perform model-checking on a forest $\mathbb{F}$ of depth $f(d)$, satisfying $||\mathbb{F}|| \leq f(d) \cdot ||A||$

Using the previous algorithm, this can be done in time $f(\varphi, d) \cdot ||A||$.

Bounded expansion graphs

Definition

An r-shallow minor H of G is a graph obtained from G by contracting connected subgraphs of radius at most r and removing some edges and vertices.

$\mathcal{C}\nabla r$: the set of shallow minors of graphs of $\mathcal{C}$.

Definition

A class of graphs $\mathcal{C}$ has **bounded expansion** if at every distance the average number of neighbours is bounded.

Bounded expansion graphs

Definition

An r-shallow minor H of G is a graph obtained from G by contracting connected subgraphs of radius at most r and removing some edges and vertices.

$\mathcal{C}\nabla r$: the set of shallow minors of graphs of $\mathcal{C}$.

Definition

A class of graphs $\mathcal{C}$ has **bounded expansion** if, for every $r \geq 0$,

$$\sup_{G \in \mathcal{C}\nabla r} \frac{|E(G)|}{|V(G)|} < \infty$$

Nowhere dense graphs

Definition

An r-shallow minor H of G is a graph obtained from G by contracting connected subgraphs of radius at most r and removing some edges and vertices.

$\mathcal{C}\nabla r$: the set of shallow minors of graphs of $\mathcal{C}$.

Definition

A class of graphs $\mathcal{C}$ is **nowhere dense** if at every distance there is no large clique.

Nowhere dense graphs

Definition

An r-shallow minor H of G is a graph obtained from G by contracting connected subgraphs of radius at most r and removing some edges and vertices.

$\mathcal{C}\nabla r$: the set of shallow minors of graphs of $\mathcal{C}$.

Definition

A class of graphs $\mathcal{C}$ is **nowhere dense** if, for every $r \geq 0$,

$$\sup_{G \in \mathcal{C}\nabla r} \omega(G) < \infty$$

Definition

An r-shallow minor H of G is a graph obtained from G by contracting connected subgraphs of radius at most r and removing some edges and vertices.

$\mathcal{C}\nabla r$: the set of shallow minors of graphs of $\mathcal{C}$.

Definition

A class of graphs $\mathcal{C}$ is **nowhere dense** if, for every $r \geq 0$,

$$\sup_{G \in \mathcal{C}\nabla r} \omega(G) < \infty$$

A class $\mathcal{C}$ is:

- nowhere dense when $\lim_{r \to +\infty} \sup_{G \in \mathcal{C}\nabla r} \frac{\log |E(G)|}{\log |V(G)|} = 1$
- somewhere dense when $\lim_{r \to +\infty} \sup_{G \in \mathcal{C}\nabla r} \frac{\log |E(G)|}{\log |V(G)|} = 2$

Polynomial expansion

Definition

A graph class $\mathcal{C}$ has bounded expansion if at every distance the average amount of neighbours is bounded.

Polynomial expansion

Definition

A graph class $\mathcal{C}$ has **polynomial** expansion if at every distance the average amount of neighbours is bounded by **a polynomial in the distance**.

Polynomial expansion

Definition

A graph class $\mathcal{C}$ has **polynomial** expansion if there is a polynomial function p s.t. for every $r \geq 0$,

$$\sup_{G \in \mathcal{C} \nabla r} \frac{|E(G)|}{|V(G)|} \leq p(r)$$

Polynomial expansion

Definition

A graph class $\mathcal{C}$ has **polynomial** expansion if there is a polynomial function p s.t. for every $r \geq 0$,

$$\sup_{G \in \mathcal{C} \nabla r} \frac{|E(G)|}{|V(G)|} \leq p(r)$$

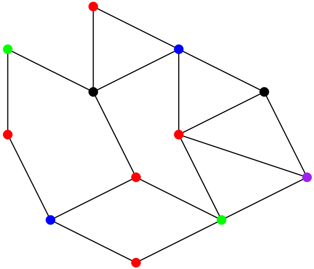
A monotone class $\mathcal{C}$ has polynomial expansion if and only if it has (balanced) separators of size $O(|V|^{1-\delta})$ for some $\delta > 0$.

Treedepth Coloring

Lemma

For $\mathcal{C}$ a class of bounded expansion and $p \in \mathbb{N}$, there exists $M = f(\mathcal{C}, p)$ such that for any $G \in \mathcal{C}$, in linear time, we can find a coloring of G using at most M colors such that any subgraph induced by at most p color classes has treedepth at most p.

Such a coloring is called a k-treedepth coloring.

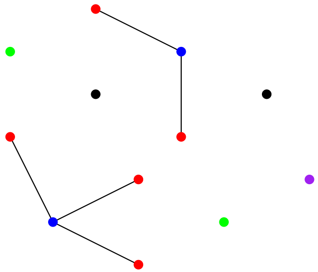


Treedepth Coloring

Lemma

For $\mathcal{C}$ a class of bounded expansion and $p \in \mathbb{N}$, there exists $M = f(\mathcal{C}, p)$ such that for any $G \in \mathcal{C}$, in linear time, we can find a coloring of G using at most M colors such that any subgraph induced by at most p color classes has treedepth at most p.

Such a coloring is called a k-treedepth coloring.

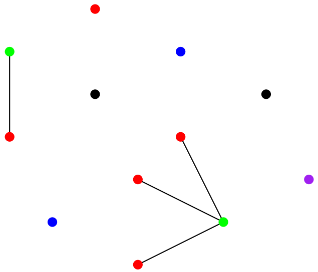


Treedepth Coloring

Lemma

For $\mathcal{C}$ a class of bounded expansion and $p \in \mathbb{N}$, there exists $M = f(\mathcal{C}, p)$ such that for any $G \in \mathcal{C}$, in linear time, we can find a coloring of G using at most M colors such that any subgraph induced by at most p color classes has treedepth at most p.

Such a coloring is called a k-treedepth coloring.



Hierarchy of sparse classes

Polynomial expansion → PTAS for IS,DS

↑

Bounded expansion → $O(1)$-approx for IS,DS

↑

Nowhere dense → $O(n^{\varepsilon})$-approx for IS,DS

Another induction on the formula

We now consider a structure $\mathbb{A}$ belonging to a class $\mathcal{C}$ of bounded expansion: the class of Gaifman graphs has bounded expansion.

We once again extend our algorithm to this setting.

Another induction on the formula

Induction on the input formula $\varphi \to$ quantifier-free $\widehat{\varphi}$.

The cases of atomic formulae, negation, and disjunction are easy, we focus on quantifiers.

Another induction on the formula

Induction on the input formula $\varphi \to$ quantifier-free $\widehat{\varphi}$.

The cases of atomic formulae, negation, and disjunction are easy, we focus on quantifiers.

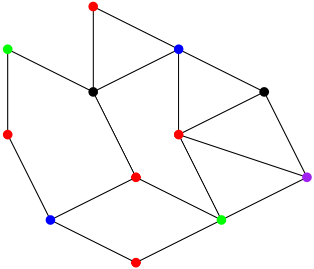
This will be done by using the previous algorithm as a **black box**!

Treedepth Coloring

Lemma

For $\mathcal{C}$ a class of bounded expansion and $p \in \mathbb{N}$, there exists $M = f(\mathcal{C}, p)$ such that for any $G \in \mathcal{C}$, in linear time, we can find a coloring of G using at most M colors such that any subgraph induced by at most p color classes has treedepth at most p.

Such a coloring is called a k-treedepth coloring.



Concluding

Consider a formula $\exists \overline{y}.\psi(\overline{x}, \overline{y})$, and let $k = |\overline{x}| + |\overline{y}|$.
In linear time, we compute a k-treedepth coloring λ of $G(\mathbb{A})$.

Check ψ: guess colors of the variables in satisfying assignments.

Concluding

Consider a formula $\exists \overline{y}.\psi(\overline{x}, \overline{y})$, and let $k = |\overline{x}| + |\overline{y}|$.
In linear time, we compute a k-treedepth coloring λ of $G(\mathbb{A})$.

Check ψ: guess colors of the variables in satisfying assignments.

For each Γ (subset of k colors): compute quantifier-free ψ^{Γ}.

Concluding

Consider a formula $\exists \overline{y}.\psi(\overline{x}, \overline{y})$, and let $k = |\overline{x}| + |\overline{y}|$.
In linear time, we compute a k-treedepth coloring λ of $G(\mathbb{A})$.

Check ψ: guess colors of the variables in satisfying assignments.

For each Γ (subset of k colors): compute quantifier-free ψ^Γ.

$$\widehat{\varphi}(\overline{x}) = \bigvee_{\Gamma} \left(\lambda(\overline{x}) \subseteq \Gamma \wedge \psi^\Gamma(\overline{x})\right)$$

Concluding

Consider a formula $\exists \overline{y}.\psi(\overline{x},\overline{y})$, and let $k = |\overline{x}| + |\overline{y}|$.
In linear time, we compute a k-treedepth coloring λ of $G(\mathbb{A})$.

Check ψ: guess colors of the variables in satisfying assignments.

For each Γ (subset of k colors): compute quantifier-free ψ^{Γ}.

$$\widehat{\varphi}(\overline{x}) = \bigvee_{\Gamma} \left(\lambda(\overline{x}) \subseteq \Gamma \wedge \psi^{\Gamma}(\overline{x})\right)$$

The total running time is $f(\mathcal{C},\varphi) \cdot ||\mathbb{A}||$

Limits of previous algorithm

Lemma

For $\mathcal{C}$ a nowhere dense class, $\varepsilon > 0$, and $p \in \mathbb{N}$, there exists $M = f(\mathcal{C}, p, \varepsilon)$ such that for any $G \in \mathcal{C}$, in time $f(\mathcal{C}, r, \varepsilon) \cdot |G|^{1+\varepsilon}$, we can find a coloring of G using at most $M \cdot |G|^{\varepsilon}$ colors such that any subgraph induced by at most p color classes has treedepth at most p.

Repeatedly guessing colors now gives an algorithm running in time

$$f(\mathcal{C}, \varphi, \varepsilon) \cdot |G|^{1+\varepsilon f(\varphi)}$$

Improved algorithm

Theorem (Grohe, Kreutzer, Siebertz)

For every nowhere dense class $\mathcal{C}$, and every $\varepsilon > 0$, any first order sentence φ, given a structure $\mathbb{A} \in \mathcal{C}$, testing φ in $\mathbb{A}$ can be done in time $f(\mathcal{C}, \varphi, \varepsilon) \cdot |\mathbb{A}|^{1+\varepsilon}$.

The strategy is to compute a normal form that does not increase the quantifier rank, and which consists of local formulae and independence predicates.

Optimality for monotone classes

We saw that FO model checking is FPT in nowhere dense classes.

Theorem

FO model checking is AW[$*$]-hard on the class of all graphs.

If a class of graphs is somewhere dense, it contains all graphs as t-subdivisions for some constant t.

Hereditary classes

In a monotone class (closed by subgraph), containing a clique is sufficient to contain every graph.

Hereditary classes

In a monotone class (closed by subgraph), containing a clique is sufficient to contain every graph.

In a hereditary class (closed by **induced** subgraph), there are simple classes that contain cliques.

$\mathcal{C} = \{K_k : k > 0\}$ is hereditary.

Sparsity and games

Sparse classes can be characterized via games:

- Cops and robber strategies for a robber of bounded speed
- Splitter game

Sparsity and games

Sparse classes can be characterized via games:

- Cops and robber strategies for a robber of bounded speed
- Splitter game

Both games have been generalized to the hereditary setting by replacing vertex removal by *flips*.

Flipping a pair of sets of vertices A, B consists in complementing the set $E(A, B)$ of edges with one endpoint in A and one endpoint in B.

Extensions to the dense setting