APPROXIMATE FIRST-ORDER COUNTING QUERIES ON SPARSE AND DENSE GRAPHS

Jan Dreier
Vienna University of Technology

joint work with Peter Rossmanith

March 16, 2021
“All problems expressible in a certain logic can be solved efficiently on certain graphs”
“All problems expressible in a certain logic can be solved efficiently on certain graphs”
Algorithmic Meta-Theorems

“All problems expressible in a certain logic can be solved efficiently on certain graphs”

MSO on treewidth

FO on sparse graphs
Algorithmic Meta-Theorems

“All problems expressible in a certain logic can be solved efficiently on certain graphs”

- MSO on treewidth
- FO on sparse graphs
- FO(\{>0\}) for approximation on sparse graphs
“All problems expressible in a certain logic can be solved efficiently on certain graphs”

- FO on sparse graphs
- FO(\{>0\}) for approximation on sparse graphs
- MSO on cliquewidth
“All problems expressible in a certain logic can be solved efficiently on certain graphs”
Many problems can be expressed in first-order (FO) logic.
Many problems can be expressed in first-order (FO) logic.

- independent set of size $k$:

$$\exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not \sim x_j \land x_i \neq x_j$$
Many problems can be expressed in first-order (FO) logic.

- independent set of size $k$:
  $$\exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \neq x_j$$

- dominating set of size $k$:
  $$\exists x_1 \ldots \exists x_k \forall y \bigvee_i y \sim x_i \lor y = x_i$$
Many problems can be expressed in first-order (FO) logic.

- independent set of size $k$:
  \[ \exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \land x_i \neq x_j \]

- dominating set of size $k$:
  \[ \exists x_1 \ldots \exists x_k \forall y \bigvee_i y \sim x_i \lor y = x_i \]

- basic database queries
Many problems can be expressed in first-order (FO) logic.

- independent set of size $k$:
  \[ \exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \land x_i \neq x_j \]

- dominating set of size $k$:
  \[ \exists x_1 \ldots \exists x_k \forall y \bigvee_i y \sim x_i \lor y = x_i \]

- basic database queries

Best algorithms on general graphs: $n^{O(k)}$
$$\phi = \exists x_1 \ldots \exists x_k [ \ldots ]$$

length depends on $k$-independent set

$k$-dominating set

...
Model-Checking

$\phi = \exists x_1 \ldots \exists x_k \ldots$

$k$-independent set

$k$-dominating set

... logic

model-checking

**MC($G, L$)**

**Input:** A graph $G \in G$ and a sentence $\varphi \in L$

**Parameter:** $|\varphi|$

**Problem:** Is $\varphi$ true in $G$?

**Goal:** linear FPT run time $f(|\varphi|)n$
Model-Checking

$k$-independent set

$k$-dominating set

... logic

logic

... model-checking

$\varphi = \exists x_1 \ldots \exists x_k [...]$

length depends on $k$

$f(|\varphi|)n$

algorithm

$\mathbf{MC}(G, L)$

Input: A graph $G \in \mathcal{G}$ and a sentence $\varphi \in \mathcal{L}$

Parameter: $|\varphi|$

Problem: Is $\varphi$ true in $G$?

Goal: linear FPT run time $f(|\varphi|)n$
If $\mathcal{G}$ has bounded treewidth then $\text{MC}(\mathcal{G}, \text{MSO}) \in \text{FPT}$.

[Courcelle 1990]
Some Sparse Graph Classes

If $\mathcal{G}$ has bounded treewidth then $\text{MC}(\mathcal{G}, \text{MSO}) \in \text{FPT}$.

[Courcelle 1990]
Some Sparse Graph Classes

If $\mathcal{G}$ has bounded treewidth then $MC(\mathcal{G}, \text{MSO}) \in \text{FPT}$.

[Courcelle 1990]
Some Sparse Graph Classes

If $\mathcal{G}$ has bounded treewidth then $MC(\mathcal{G}, \text{MSO}) \in \text{FPT}$.

[Courcelle 1990]

If $\mathcal{G}$ is nowhere dense then $MC(\mathcal{G}, \text{FO}) \in \text{FPT}$.

[Grohe, Kreutzer, Sieberz 2011]
Some Sparse Graph Classes

If $G$ has bounded treewidth then $\text{MC}(G, \text{MSO}) \in \text{FPT}$.

[Courcelle 1990]

If $G$ is nowhere dense then $\text{MC}(G, \text{FO}) \in \text{FPT}$.

[Grohe, Kreutzer, Sieberz 2011]
Bounded Expansion — Minors

\[
\frac{|E|}{|V|} \leq C
\]

for every graph in the graph class.
Bounded Expansion — Minors

\[ \frac{|E|}{|V|} \leq 2 \]

for every graph in the graph class.
Bounded Expansion — Minors

radius $r$

$r$— shallow topological minor
Bounded Expansion — Minors

radius $r$

$r$— shallow topological minor
Bounded Expansion — Minors

radius $r$

$r$—shallow topological minor
Bounded Expansion — Minors

For every $r$-shallow minor of every graph in the graph class.

$$\frac{|E|}{|V|} \leq f(r)$$

for every $r$-shallow minor of every graph in the graph class.
Bounded Expansion — Augmentations

Possible to do these augmentations such that after $r$ augmentations, outdegree at most $f(r)$.
Bounded Expansion — Augmentations

Possible to do these augmentations such that after $r$ augmentations, outdegree at most $f(r)$.

0th Augmentation

orient edges

max outdegree $= 2$
Possible to do these augmentations such that after $r$ augmentations, outdegree at most $f(r)$.

$1^{st}$ Augmentation

max outdegree = 2

orient edges

transitive rule

or
Possible to do these augmentations such that after $r$ augmentations, outdegree at most $f(r)$.

$max\ \text{outdegree} = 3$

orient edges

transitive rule

$1^{st} \ \text{Augmentation}$

max outdegree $\geq 3$
Bounded Expansion — Augmentations

Possible to do these augmentations such that after $r$ augmentations, outdegree at most $f(r)$.

1st Augmentation

max outdegree = 3

orient edges

or

transitive rule

or

fraternal rule

6
Possible to do these augmentations such that after \( r \) augmentations, outdegree at most \( f(r) \).
Possible to do these augmentations such that after $r$ augmentations, outdegree at most $f(r)$. 

$max \text{ outdegree} = 3$

$2^\text{nd} \text{ Augmentation}$

orient edges

or

transitive rule

fraternal rule

$max \text{ outdegree} = 3$
Approximate Counting Queries
### Partial Dominating Set

**Input:** A graph $G$ and $k, m \in \mathbb{N}$  
**Parameter:** $k$  
**Problem:** Are there $k$ vertices dominating $m$ vertices?
### Partial Dominating Set

**Input:** A graph $G$ and $k, m \in \mathbb{N}$

**Parameter:** $k$

**Problem:** Are there $k$ vertices dominating $m$ vertices?

Cannot be expressed in first-order logic (requires $\exists y_1 \ldots \exists y_m$).
**Partial Dominating Set**

**Input:** A graph $G$ and $k, m \in \mathbb{N}$

**Parameter:** $k$

**Problem:** Are there $k$ vertices dominating $m$ vertices?

Cannot be expressed in first-order logic (requires $\exists y_1 \ldots \exists y_m$).

Can be solved on $H$-minor free graphs in time $(g(H)^k)^k n^{O(1)}$.

[Amini, Fomin, Saurabh, 2008]

Can be solved on apex-minor-free graphs in time $2^{\sqrt{k}} n^{O(1)}$.

[Fomin, Lokshtanov, Raman, Saurabh, 2011]


[Golovach, Villanger 2008]
### Partial Dominating Set

**Input:** A graph $G$ and $k, m \in \mathbb{N}$

**Parameter:** $k$

**Problem:** Are there $k$ vertices dominating $m$ vertices?

\[
\exists x_1 \ldots \exists x_k \# y (\bigvee_i y \sim x_i \lor y = x_i) \geq m
\]

Length of formula depends only on $k$ (and not on $m$)

\[
\text{FO} \{ > 0 \} = \text{FO} \text{ + “there are at least/most } m \in \mathbb{N} \text{ elements”}
\]
**Partial Dominating Set**

*Input:* A graph $G$ and $k, m \in \mathbb{N}$

*Parameter:* $k$

*Problem:* Are there $k$ vertices dominating $m$ vertices?

---

$\text{FO}(\{>0\}) = \text{FO} + \text{“there are at least/most } m \in \mathbb{N} \text{ elements”}$

$$\exists x_1 \ldots \exists x_k \#y \left( \bigvee_i y \sim x_i \lor y = x_i \right) \geq m$$
Partial Dominating Set

**Input:** A graph $G$ and $k, m \in \mathbb{N}$

**Parameter:** $k$

**Problem:** Are there $k$ vertices dominating $m$ vertices?

$\text{FO}(<0>) = \text{FO} + \text{"there are at least/most } m \in \mathbb{N} \text{ elements"}$

$$\exists x_1 \ldots \exists x_k \# y \left( \bigvee_{i} y \sim x_i \lor y = x_i \right) \geq m$$

Length of formula depends only on $k$ (and not on $m$)
Definition of FO(\{> 0\})

built recursively using
- the rules of FO
- \#y \varphi \geq m for every \(m \in \mathbb{N}\) and FO(\{> 0\}) formula \(\varphi\)

Example 1: PARTIAL DOMINATING SET

\[ \exists x_1 \ldots \exists x_k \#y \left( \bigvee_{i} y \sim x_i \land y = x_i \right) \geq m \]

Example 2: \(h\)-Index

\[ \#\text{mypaper} \left( \#\text{otherpaper cite(otherpaper, mypaper)} \geq h \right) \geq h \]
If $\mathcal{G}$ has bounded degree then $\text{MC}(\mathcal{G}, \text{FOC}) \in \text{FPT}$.  
[Kuske, Schweikardt 2017]
If $\mathcal{G}$ has bounded degree then $\text{MC}(\mathcal{G}, \text{FOC}) \in \text{FPT}$.  
[Kuske, Schweikardt 2017]

$\text{MC}(\mathcal{G}, \text{FO}(\{>0\}))$ is $\text{AW}[\ast]$-hard on trees.  
similar to [Grohe, Schweikardt 2018]
Bad News

contains k-clique

\[ \Leftrightarrow \]

satisfies FO(\{\text{>0}\}) formula
contains k-clique \iff satisfies FO(\{\geq 0\}) formula
contains k-clique

satisfies FO(\{\geq 0\}) formula

\[
\begin{array}{c}
\text{i} \\
\text{j} \\
\text{k}
\end{array}
\]

\[
\begin{array}{c}
\text{i} \\
\text{n-j} \\
\text{n-k} \\
\text{n-l}
\end{array}
\]
Bad News

contains k-clique satisfies FO(>0) formula
Are there \( k \) vertices dominating at least \( m = 5000 \) vertices?
Are there $k$ vertices dominating at least $m = 4983$ vertices?

\[ x_1 \ldots x_k \geq 4983 \]
Are there $k$ vertices dominating at least $m = 5017$ vertices?
Are there \( k \) vertices dominating at least \( m = 5017 \) vertices?

Let \( \varepsilon > 0 \). A formula \( \varphi \) is \( \varepsilon \)-unstable on a graph \( G \) if scaling the counting literals by \( (1 \pm \varepsilon) \) changes whether \( \varphi \) is true in \( G \).
Theorem

Let $G$ be a graph class with bounded expansion and $\varepsilon > 0$. 
Theorem

Let $\mathcal{G}$ be a graph class with bounded expansion and $\varepsilon > 0$. There exists an algorithm which takes $G \in \mathcal{G}$, $\varphi \in \text{FO}(\{>0\})$, runs in time $f(|\varphi|)n$ and returns $\smiley$, $\sad$, or $\sunglass$. 
Approximate Model-Checking

Theorem

Let $G$ be a graph class with bounded expansion and $\varepsilon > 0$. There exists an algorithm which takes $G \in \mathcal{G}$, $\varphi \in \text{FO}(\{>0\})$, runs in time $f(|\varphi|)n$ and returns \(\smiley\), \(\frown\), or \(\moy\).

- If \(\smiley\) then $\varphi$ is true on $G$. 

- If \(\frown\) then $\varphi$ is false on $G$.

- If \(\moy\) then $\varphi$ is $\varepsilon$-unstable on $G$. 

Theorem

Let $\mathcal{G}$ be a graph class with bounded expansion and $\varepsilon > 0$. There exists an algorithm which takes $G \in \mathcal{G}$, $\varphi \in \text{FO}(\{\geq 0\})$, runs in time $f(|\varphi|)n$ and returns \(\bigcirc\), \(\bigcirc\), or \(\bigcirc\).

- If \(\bigcirc\) then $\varphi$ is true on $G$.
- If \(\bigcirc\) then $\varphi$ is false on $G$. 
Theorem

Let $\mathcal{G}$ be a graph class with bounded expansion and $\varepsilon > 0$. There exists an algorithm which takes $G \in \mathcal{G}$, $\varphi \in \text{FO}(\{\geq 0\})$, runs in time $f(|\varphi|)n$ and returns $\bigcirc$, $\bigtimes$, or $\bigotimes$.

- If $\bigcirc$ then $\varphi$ is true on $G$.
- If $\bigtimes$ then $\varphi$ is false on $G$.
- If $\bigotimes$ then $\varphi$ is $\varepsilon$-unstable on $G$. 
Partial Dominating Set: \( \exists x_1 \ldots \exists x_k \# y \left( \bigvee_i y \sim x_i \land y = x_i \right) \geq m \)
**Approximate Model-Checking**

**Partial Dominating Set:** \( \exists x_1 \ldots \exists x_k \# y (\bigvee_i y \sim x_i \land y = x_i) \geq m \)

There exists a set dominating \( \geq (1 + \varepsilon)m \) vertices.
**Approximate Model-Checking**

**PARTIAL DOMINATING SET:** \( \exists x_1 \ldots \exists x_k \# y \left( \bigvee_i y \sim x_i \land y = x_i \right) \geq m \)

There exists a set dominating \( \geq (1 + \varepsilon)m \) vertices.

All sets dominate \( < (1 - \varepsilon)m \) vertices.
**Partial Dominating Set:** $\exists x_1 \ldots \exists x_k \# y \left( \bigvee_i y \sim x_i \land y = x_i \right) \geq m$

There exists a set dominating $\geq (1 + \varepsilon)m$ vertices.

All sets dominate $< (1 + \varepsilon)m$ vertices and there exists a set dominating $\geq (1 - \varepsilon)m$ vertices.

All sets dominate $< (1 - \varepsilon)m$ vertices.
Theorem

\textsc{PartialDominatingSet} can be solved in time $f(k)n$ on graph classes with bounded expansion.
Theorem

**PARTIALDOMINATINGSET** can be solved in time $f(k)n$ on graph classes with bounded expansion.

This holds for all problems of the form

$$\exists x_1 \ldots \exists x_k \#y \varphi(y, x_1 \ldots x_k) \geq m. \in \text{FO}$$
How about extensions of FO(\{> 0\})?

FO(\{> 0\}) allows comparing $\#y$ and $m \in \mathbb{N}$.

**Theorem**

Approximate model-checking becomes hard on trees if also allow one of the following:
How about extensions of FO(\{>0\})?

FO(\{>0\}) allows comparing \#y and \(m \in \mathbb{N}\).

**Theorem**

Approximate model-checking becomes hard on trees if also allow one of the following:

- comparing \#y and \#z \hspace{1cm} \text{(e.g., } \#y \varphi > \#z \psi \text{)}
How about extensions of $\text{FO}(\{>0\})$?

$\text{FO}(\{>0\})$ allows comparing $\#y$ and $m \in \mathbb{N}$.

**Theorem**

Approximate model-checking becomes hard on trees if also allow one of the following:

- comparing $\#y$ and $\#z$  
  (e.g., $\#y \varphi > \#z \psi$)
- counting tuples $\#yz$  
  (e.g., $\#yz \varphi > m$)
How about extensions of $\text{FO}(\{\geq 0\})$?

$\text{FO}(\{\geq 0\})$ allows comparing $\#y$ and $m \in \mathbb{N}$.

**Theorem**

Approximate model-checking becomes hard on trees if also allow one of the following:

- comparing $\#y$ and $\#z$  
  (e.g., $\#y \varphi > \#z \psi$)
- counting tuples $\#yz$  
  (e.g., $\#yz \varphi > m$)
- multiplying of counting terms  
  (e.g., $\#y \varphi \cdot \#z \psi > m$)
How about extensions of FO(\{ > 0 \})?

FO(\{ > 0 \}) allows comparing \#y and \( m \in \mathbb{N} \).

**Theorem**

Approximate model-checking becomes hard on trees if also allow one of the following:

- comparing \#y and \#z  
  (e.g., \#y \varphi > \#z \psi)
- counting tuples \#yz  
  (e.g., \#yz \varphi > m)
- multiplying of counting terms  
  (e.g., \#y \varphi \cdot \#z \psi > m)
- subtraction of counting terms  
  (e.g., \#y \varphi - \#z \psi > m)
Summary

$\text{FO}\{>0\}$ is

- hard to solve exactly on trees,
Summary

FO(\{>0\}) is

- hard to solve exactly on trees,
- possible to approximate on bounded expansion.
Summary

$\text{FO}(\{ > 0 \})$ is

- hard to solve exactly on trees,
- possible to approximate on bounded expansion.

Slight extensions of $\text{FO}(\{ > 0 \})$ are

- hard to approximate on trees.
**Summary**

**FO(\{>0\})** is

- hard to solve exactly on trees,
- possible to approximate on bounded expansion.

Slight extensions of **FO(\{>0\})** are

- hard to approximate on trees.

$\Rightarrow$ **FO(\{>0\})** seems like “the right logic” for approximation on sparse graphs
Can we generalize our results to nowhere dense graph classes?
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \]

quantifier-free FO \( \phi(x_1,x_2,x_3) \)
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \left( \right) \right) \]
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \left( m_3 \leq \#x_3 \right) \right) \]
Proof — Quantifier Elimination

We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \left( m_3 \leq \#x_3 \left[ \varphi(x_1x_2x_3) \right] \right) \right) \]
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \left( m_3 \leq \#x_3 \phi(x_1, x_2, x_3) \right) \right) \]
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \quad \varphi'(x_1x_2) \right) \]
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \varphi'(x_1x_2) \right) \]

replace with quantifier-free FO

quantifier-free FO
We want to gradually simplify this formula.

\[ m_1 \leq \#x_1 \quad \{ \varphi''(x_1) \} \]
Proof — Quantifier Elimination

We want to gradually simplify this formula.

$$m_1 \leq \#x_1 \approx \varphi''(x_1)$$

replace with quantifier-free FO
We want to gradually simplify this formula.
Proof — Functional Representation

\[ w \quad \rightarrow \quad v \]

\[ w \quad \rightarrow \quad \]

\[ v \quad \rightarrow \quad \]
Proof — Functional Representation

bounded outdegree

$w$ $v$

$w$ $v$
Proof — Functional Representation

\[ f_1(v) = w \]
Proof — Functional Representation

\[ x_1 \sim x_2 \]

\[ f_1(v) = w \]

\[ \bigvee_i f_i(x_1) = x_2 \lor f_i(x_2) = x_1 \]
\[
\#y \left( f_1(y) = x_1 \land f_2(y) = x_2 \right) \geq m
\]
Proof — Equalities Only

fraternal rule:

\[ \Rightarrow \quad \text{or} \quad \]}

\[
\#y \left( \begin{array}{l}
    f_1(y) = x_1 \land \\
    f_2(y) = x_2
\end{array} \right) \geq m
\]
fraternal rule:
\[ \iff \quad \lor \iff \]

\[
\#y \left( f_1(y) = x_1 \land f_2(y) = x_2 \right) \geq m
\]
\[ \bigvee_{h} \left( h(x_1) = x_2 \land \right. \]
\[ \#y \left( f_1(y) = x_1 \land f_2(y) = x_2 \right) \]
\[ ) \geq m \]
\[ \left. \right) \]
\[
\bigvee_{h} \left( h(x_1) = x_2 \land \\
\#y \left( \\
f_1(y) = x_1 \land \\
\underline{f_2(y) = x_2} \land \\
f_2(y) = h(x_1) \right) \geq m \right)
\]
Proof — Equalities Only

\[ \bigvee_{h} \left( h(x_1) = x_2 \land \right. \]
\[ \left. \#y \left( f_1(y) = x_1 \land \right. \right. \]
\[ \left. \left. \overline{f_2(y) = x_2} \land \right. \right. \]
\[ \left. f_2(y) = h(x_1) \right) \geq m \]

\[ \land \quad \#y \geq m \]

\[ \quad f_1 \quad \overline{f_2} \]

\[ x_1 \rightarrow x_2 \quad h \]

\[ x_1 \rightarrow \quad h \]

\[ \#y \varphi(yx_1) \geq m \]
Proof — Equalities Only

\[ \bigvee_{h} \left( h(x_1) = x_2 \land \#y \left( f_1(y) = x_1 \land f_2(y) = x_2 \land f_2(y) = h(x_1) \right) \geq m \right) \]

Evaluate for every vertex

\[ \#y \varphi(yx_1) \geq m \iff R_h(x_1) \]
Proof — Equalities Only

\[ \bigvee_{h} \left( h(x_1) = x_2 \land R_h(x_1) \right) \]
\[ \#y \left( f_1(y) = x_1 \land f_2(y) = x_2 \land f_2(y) = h(x_1) \right) \geq m \]

evaluate for every vertex

\[ x_1 \xrightarrow{h} x_2 \]

\[ \#y \varphi(yx_1) \geq m \iff R_h(x_1) \]
Proof — Equalities Only

\[ \bigvee_{h} \left( h(x_1) = x_2 \land R_h(x_1) \right) \]

\[ \# y \left( f_1(y) = x_1 \land f_2(y) = x_2 \land f_2(y) = h(x_1) \right) \geq m \]

\[ x_1 \xrightarrow{h} x_2 \]

\[ \# y \geq m \]

\[ \varphi(yx_1) \geq m \iff R_h(x_1) \]

evaluate for every vertex
\[ \# y \ f_1(y) = x_1 \land \\
    f_2(y) \neq x_2 \]
\[ \#y \ f_1(y) = x_1 \wedge \]
\[ f_2(y) \neq x_2 \]
\#y \ f_1(y) = x_1 \land \\
f_2(y) \neq x_2
Proof — Inequalities

\[ \#y f_1(y) = x_1 \land f_2(y) \neq x_2 \]
Proof — Inequalities

\[ \#y \ f_1(y) = x_1 \land f_2(y) \neq x_2 \]
\[
\#y \ f_1(y) = x_1 \land f_2(y) \neq x_2
\]
Proof — Inequalities

\[ \#y \ f_1(y) = x_1 \land \ f_2(y) \neq x_2 \]

add extra edge and proceed as before
Proof — Inequalities

\#y \ f_1(y) = x_1 \land \ f_2(y) \neq x_2

still bounded outdegree!

add extra edge and proceed as before
Proof — Quantifier Elimination

Gradually simplify formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \left( m_3 \leq \#x_3 \varphi(x_1 x_2 x_3) \right) \right) \]

replace with quantifier-free FO

quantifier-free FO
Gradually simplify formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \quad \varphi'(x_1x_2) \right) \]
Gradually simplify formula.

\[ m_1 \leq \#x_1 \left( m_2 \leq \#x_2 \quad \varphi'(x_1, x_2) \right) \]

quantifier-free FO

replace with quantifier-free FO
Gradually simplify formula.

\[ m_1 \leq \#x_1 \quad \{ \varphi''(x_1) \} \]
Gradually simplify formula.

\[ m_1 \leq \#x_1 \quad \varphi''(x_1) \]

replace with quantifier-free FO
Gradually simplify formula.
Gradually simplify formula.
Dense Graphs
For graph classes $\mathcal{G}$ closed under subgraphs, FO model-checking is tractable iff $\mathcal{G}$ is nowhere dense.

[Grohe, Kreutzer, Sieberz 2011]
What dense graph classes are tractable?
What dense graph classes are tractable?

Closure under subgraphs is not a good requirement.
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
What dense graph classes are tractable?

Closure under subgraphs is not a good requirement.
What dense graph classes are tractable?

Closure under subgraphs is not a good requirement.
What dense graph classes are tractable?

Closure under subgraphs is not a good requirement.
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
- **Goal:** For graph classes \( G \) closed under induced subgraphs, FO model-checking is tractable iff [...].
Example: Complements
Example: Complements

\[ x_1 \sim x_2 \quad \text{complement} \quad x_1 \not\sim x_2 \]
Example: Complements

\[ \exists x_1 \exists x_2 \exists x_3 \]
\[ x_1 \sim x_2 \land x_2 \sim x_3 \]
\[ \land x_1 \sim x_3 \]

\[ \Leftrightarrow \]

\[ \exists x_1 \exists x_2 \exists x_3 \]
\[ x_1 \not\sim x_2 \land x_2 \not\sim x_3 \]
\[ \land x_1 \not\sim x_3 \]
Example: Fully Bipartite
Example: Fully Bipartite
Example: Fully Bipartite

\[ x \sim y \]

\[ \text{dist}(x, y) = 3 \]
Example: Fully Bipartite

\[ x \sim y \]

\[ \exists x \ \varphi \]

\[ \text{dist}(x, y) = 3 \]

\[ \exists x \ \text{blue}(x) \land \varphi \]
Example: Fully Bipartite

\[ \exists x_1 \exists x_2 \exists x_3 \]
\[ x_1 \sim x_2 \land x_2 \sim x_3 \land x_1 \sim x_3 \]

\[ \exists x_1 \exists x_2 \exists x_3 \]
\[ \text{blue}(x_1) \land \text{blue}(x_2) \land \text{blue}(x_3) \land \text{dist}(x_1, x_2) = 3 \land \text{dist}(x_2, x_3) = 3 \land \text{dist}(x_1, x_3) = 3 \]
Interpretations

\[ I = (\nu(x), \mu(x, y)) \]
Interpretations

\[ I = (\nu(x), \mu(x, y)) \]

vertices: \( \{v \mid G \models \nu(v)\} \)

edges: \( \{uv \mid G \models \mu(u, v)\} \)
Interpretations

\[ I = (\nu(x), \mu(x, y)) \]

\[ G \]

all blue vertices

\[ I(G) \]

vertices: \( \{v \mid G \models \nu(v)\} \)

edges: \( \{uv \mid G \models \mu(u, v)\} \)
Interpretations

$$I = (\nu(x), \mu(x, y))$$

vertices: \{v \mid G \models \nu(v)\}

edges: \{uv \mid G \models \mu(u, v)\}
A graph class $\mathcal{G}$ has *structurally property* $X$ if there exists
A graph class $\mathcal{G}$ has \textit{structurally property} $X$ if there exists

- a class $\mathcal{G}'$ with property $X$, 

The class of all fully bipartite graphs has \textit{structurally treewidth} 1:

- For every there is $\mathcal{G}'$ with $\mathcal{G}' = I(\mathcal{G})$. 

A graph class $\mathcal{G}$ has *structurally property* $X$ if there exists

- a class $\mathcal{G}'$ with property $X$,
- an interpretation $I = (\nu(x), \mu(x, y))$, 

The class of all fully bipartite graphs has *structurally treewidth* 1.
A graph class $\mathcal{G}$ has *structurally property* $X$ if there exists

- a class $\mathcal{G}'$ with property $X$,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$. 
A graph class $\mathcal{G}$ has \textit{structurally property X} if there exists

- a class $\mathcal{G}'$ with property X,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

The class of all fully bipartite graphs has \textit{structurally treewidth 1}:
A graph class $\mathcal{G}$ has *structurally property* $X$ if there exists

- a class $\mathcal{G}'$ with property $X$,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

The class of all fully bipartite graphs has *structurally treewidth 1*:

- The class of all \(\quad\)

  \begin{center}
  \begin{tikzpicture}
  \node[shape=circle,draw=black,fill=black] (n1) at (0,0) {};
  \node[shape=circle,draw=black,fill=black] (n2) at (1,1) {};
  \node[shape=circle,draw=black,fill=black] (n3) at (1,-1) {};
  \end{tikzpicture}
  \end{center}

  has treewidth 1
A graph class $\mathcal{G}$ has \textit{structurally property} $X$ if there exists

- a class $\mathcal{G}'$ with property $X$,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

The class of all fully bipartite graphs has \textit{structurally treewidth} 1:

- The class of all \includegraphics[width=0.2\textwidth]{example_graph} has treewidth 1
- For every \includegraphics[width=0.3\textwidth]{example_graph} there is \includegraphics[width=0.2\textwidth]{example_graph} with $G = I(G')$. 
Nowhere Dense: Grohe, Kreutzer, Sieberz 2011
Structurally Bounded Degree: Gajarský, Hlinenỳ, Obdržálek, Lokshtanov, Ramanujan 2016
Structurally Bounded Expansion

$I(G)$
Structurally Bounded Expansion

Bounded Expansion $G$

$\rightarrow$

Structurally Bounded Expansion $I(G)$
Structurally Bounded Expansion

Bounded Expansion

$G$

Structurally Bounded Expansion

$I(G')$

$I = (\nu(x), \mu(x, y))$

$x \sim y \rightarrow \mu(x, y)$

$\exists x \rightarrow \exists x \nu(x) \land$

$\varphi'$

$\varphi$
Structurally Bounded Expansion

\[ I = (\nu(x), \mu(x, y)) \]

\[ x \sim y \rightarrow \mu(x, y) \]
\[ \exists x \rightarrow \exists x \nu(x) \land \]

MC-algorithm
Structurally Bounded Expansion

\[ G \]

Bounded Expansion

\[ I(G) \]

Structurally Bounded Expansion

\[ I = (\nu(x), \mu(x, y)) \]

\[ x \sim y \rightarrow \mu(x, y) \]
\[ \exists x \rightarrow \exists \nu(x) \land \]

\[ \varphi' \]

MC-algorithm
Structurally Bounded Expansion

Bounded Expansion $G$

Structurally Bounded Expansion $I(G)$

$\varphi'$

MC-algorithm

$\varphi$

$I = (\nu(x), \mu(x, y))$

$x \sim y \rightarrow \mu(x, y)$

$\exists x \rightarrow \exists x \nu(x) \wedge$

hard to find
Structurally Bounded Degree

Degree 3

$G$

Structurally
Degree 3

$I(G)$

NP-complete to find preimage
Structurally Bounded Degree

Degree 3

$G$

Degree $k$

Structurally
Degree 3

$I(G)$

$I$

$I'$

polynomially computable
Structurally Bounded Degree

Degree $d$

$G$

Degree $f(d)$

Structurally Degree $d$

$I(G)$

$I$

$I'$
Structurally Bounded Degree

Degree $d$

$G$

Degree $f(d)$

Structurally
Degree $d$

$I(G)$

$I$

$I'$

computable in FPT time
Model-Checking in Sparse and Dense Classes

Sparse

- Somewhere dense
- Nowhere dense
- Bounded expansion
- (Top.) minor free
- Planar
- Bounded treewidth
- Bounded degree

Dense

- Structurally nowhere dense
- Structurally bounded expansion
- Structurally bounded degree

Interpretation

Nowhere Dense: Grohe, Kreutzer, Sieberz 2011
Structurally Bounded Degree: Gajarský, Hlinenỳ, Obdržálek, Lokshtanov, Ramanujan 2016
Big Question

Bounded Expansion

\[ G \]

Structurally Bounded Expansion

\[ I(G) \]
Big Question

Bounded Expansion

$G$

Bounded Expansion

Structurally Bounded Expansion

$I(G)$

$I$

$I'$
Big Question

Bounded Expansion

\[ G \]

Bounded Expansion

Lacon Decomposition

\[ I(G) \]

Structurally Bounded Expansion
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output

1 0 0 1
Lacon Decompositions

Lacon Decomposition

Output

37
Lacon Decompositions

Lacon Decomposition

Output

has lacon decomposition with property X

estructurally property X
Lacon Decompositions

Lacon Decomposition

Output

structurally

property X

has lacon decomposition

with property X

has lacon decomposition with property X

structurally property X
Lacon Decompositions

Lacon Decomposition

Output

structurally
treewidth 1
treewidth 3

treewidth 3

structurally
treewidth 1
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output

38
Big Question

Structurally Bounded Expansion
Big Question

has Lacon Decomposition with Bounded Expansion

Structurally Bounded Expansion
Big Question

has Lacon Decomposition with Bounded Expansion

Can we compute it?
That would solve the model-checking problem.
Big Question

Structurally Nowhere Dense
Big Question

Has nowhere dense Lacon decomposition?

Structurally nowhere dense
Thanks!