Complex Networks
meet Sparsity
Solving Problems Definable in First-Order Logic

Jan Dreier, Philipp Kuinke, Peter Rossmanith

May 25 2021

RWTH Aachen University, TU Wien
Algorithmic Meta-Theorems

“All problems expressible in a certain logic can be solved efficiently on certain graph classes.”
Algorithmic Meta-Theorems

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

\[ C = \{ G_1, \ldots, G_\infty \} \]
Motivation

real world networks
Motivation

real world networks \[\rightarrow\] network science \[\rightarrow\] random graph models
Motivation

- Real world networks
- Random graph models
- Network science
- Algorithmic meta-theorem

∀x ∃y
Motivation

real world networks → network science → random graph models → practical algorithms? → engineering → algorithmic meta-theorem

int main(int argc, char *argv[]) {...}
Motivation

- Real world networks
- Random graph models
- Network science
- Practical algorithms?
- Engineering
- Algorithmic meta-theorem

```c
int main(int argc, char *argv[]) {...}
```
Motivation

real world networks → network science → random graph models

network science → practical algorithms?

practical algorithms? → engineering → algorithmic meta-theorem

\[ \forall x \exists y \]
FO Model-Checking [Grohe, Kreutzer, Sieberz 2011]

First-order formulas $\varphi$ can be evaluated on sparse graphs in time $f(|\varphi|)n^{1+\varepsilon}$. 
First-order formulas $\varphi$ can be evaluated on sparse graphs in time $f(|\varphi|)n^{1+\varepsilon}$.

- Dominating set of size $k$:

$$\exists x_1 \ldots \exists x_k \forall y \bigvee_{i} y \sim x_i \lor y = x_i$$
First-order formulas $\varphi$ can be evaluated on sparse graphs in time $f(|\varphi|)n^{1+\varepsilon}$.

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

- **independent set of size $k$:**
  \[
  \exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \land x_i \neq x_j
  \]
First-order formulas $\varphi$ can be evaluated on sparse graphs in time $f(|\varphi|)n^{1+\varepsilon}$.

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

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

Best algorithms on general graphs: $n^{O(k)}$
First-order formulas \( \varphi \) can be evaluated on sparse graphs in time \( f(|\varphi|)n^{1+\varepsilon} \).

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

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

Best algorithms on general graphs: \( n^{O(k)} \)

On sparse graphs: \( f(k)n^{1+\varepsilon} \)
can be expressed:
First-Order Logic

can be expressed:

- dominating set of size $k$
First-Order Logic

can be expressed:

- dominating set of size $k$
- independent set of size $k$

fundamentals of SQL database queries on general graphs:
$n^O(k)$, on sparse graphs:
$f(k) n^{1+\varepsilon}$

are two vertices in the same connected component?
do two vertices have distance at most $\log(n)$?
is something acyclic?
First-Order Logic

can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
- fundamentals of SQL database queries
First-Order Logic

can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
- fundamentals of SQL database queries

on general graphs: $n^{O(k)}$, on sparse graphs: $f(k)n^{1+\varepsilon}$
First-Order Logic can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
- fundamentals of SQL database queries

on general graphs: $n^{O(k)}$, on sparse graphs: $f(k)n^{1+\varepsilon}$

cannot be expressed:

- are two vertices in the same connected component?
- do two vertices have distance at most $\log(n)$?
- is something acyclic?
can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
- fundamentals of SQL database queries

on general graphs: $n^{O(k)}$, on sparse graphs: $f(k) n^{1+\varepsilon}$

cannot be expressed:

- are two vertices in the same connected component?
First-Order Logic

can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
- fundamentals of SQL database queries

on general graphs: $n^{O(k)}$, on sparse graphs: $f(k)n^{1+\varepsilon}$

cannot be expressed:

- are two vertices in the same connected component?
- do two vertices have distance at most $\log(n)$?
First-Order Logic

can be expressed:

- dominating set of size $k$
- independent set of size $k$
- motif counting of size $k$
- fundamentals of SQL database queries

on general graphs: $n^{O(k)}$, on sparse graphs: $f(k)n^{1+\varepsilon}$

cannot be expressed:

- are two vertices in the same connected component?
- do two vertices have distance at most $\log(n)$?
- is something acyclic?
First-Order Logic

can be expressed:

- dominating set of size \( k \)
- independent set of size \( k \)
- motif counting of size \( k \)
- fundamentals of SQL database queries

on general graphs: \( n^{O(k)} \), on sparse graphs: \( f(k) n^{1+\varepsilon} \)

cannot be expressed:

- are two vertices in the same connected component?
- do two vertices have distance at most \( \log(n) \)?
- is something acyclic?
Sparse Graph Classes

Somewhere Dense

Nowhere Dense

Bounded Expansion

(Top.) Minor Free

Planar

Bounded Degree

Bounded Treewidth
What Are Sparse Graphs?

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

for every graph in the graph class.

In the real world, for small \( r \) (such as 1, 2, 3), sparse (w.r.t. gen. col. numbers)

[Nadara, Pilipczuk, Rabinovich, Reidl, Siebertz 2019]
What Are Sparse Graphs?

for every graph in the graph class.

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

for every graph in the graph class.

real world: for small \(r\) (such as 1, 2, 3) sparse (w.r.t. gen. col. numbers)

[Nadara, Pilipczuk, Rabinovich, Reidl, Siebertz 2019]
What Are Sparse Graphs?

radius $r$

$r$—shallow minor

[Real world: for small $r$ (such as 1, 2, 3) sparse (w.r.t. gen. col. numbers)]

[Nadara, Pilipczuk, Rabinovich, Reidl, Siebertz 2019]
What Are Sparse Graphs?

radius $r$

$r$—shallow minor

[Real-world example: for small $r$ (such as 1, 2, 3) sparse (w.r.t. general column numbers)]

[Nadara, Pilipczuk, Rabinovich, Reidl, Siebertz 2019]
What Are Sparse Graphs?

radius $r$

$r$—shallow minor

[Nadara, Pilipczuk, Rabinovich, Reidl, Siebertz 2019]
What Are Sparse Graphs?

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.

$r$-shallow minor

radius $r$
What Are Sparse Graphs?

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.

Real world: for small $r$ (such as 1,2,3) sparse (w.r.t. gen. col. numbers)

[Nadara, Pilipczuk, Rabinovich, Reidl, Siebertz 2019]
The Real World

real world networks

network science

random graph models

practical algorithms?

engineering

algorithmic meta-theorem

\[ \forall x \exists y \]
The Real World

Some central properties:

- **Skewed degree distribution**: Fraction of vertices with degree $k$ proportional to $k^{-\alpha}$ with $2 \leq \alpha \leq 3$.

- **Clustered**: If we have a common friend, we are likely friends as well.

- **Small-world property**: Everyone is close to everyone.
Some central properties:

- Skewed degree distribution: Fraction of vertices with degree $k$ proportional to $k^{-\alpha}$ with $2 \leq \alpha \leq 3$.
- Clustered: If we have a common friend, we are likely friends as well.
- Small-world property: Everyone is close to everyone.
Some central properties:

- **Skewed degree distribution**
  Fraction of vertices with degree $k$ proportional to $k^{-\alpha}$ with $2 \leq \alpha \leq 3$?
Some central properties:

- **Skewed degree distribution**
  Fraction of vertices with degree $k$ proportional to $k^{-\alpha}$ with $2 \leq \alpha \leq 3$?

- **Clustered**
  If we have a common friend we are likely friends as well
The Real World

Some central properties:

- **Skewed degree distribution**
  Fraction of vertices with degree $k$ proportional to $k^{-\alpha}$ with $2 \leq \alpha \leq 3$?

- **Clustered**
  If we have a common friend we are likely friends as well

- **Small-world property**
  Everyone is close to everyone
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Example: Preferential Attachment Model

Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.
Introduced by Barabási and Albert in 1999 to explain the structure of the world wide web.

\[
\text{expected degree} \approx \sqrt{\frac{n}{i}}
\]
Sparsity of Random Graph Models

- Somewhere Dense
- Nowhere Dense
- Bounded Expansion
- (Top.) Minor Free
- Planar
- Bounded Degree
- Bounded Treewidth

[Grohe 2001], [Farrell et. al. 2015], [Demaine et. al. 2019], [Dreier et. al. 2020]
Results
Our Result

A random graph model is $3$-power-law-bounded if (roughly speaking):

\[ \text{fraction of vertices with degree } k \text{ is } O(k^{-3}) \]

real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$

unclustered real networks: typically clustered

Theorem

Given a first-order sentence $\varphi$ and a graph $G$ sampled from a $3$-power-law-bounded model, one can decide whether $\varphi$ is true on $G$ in expected time $f(|\varphi|) n^{1+\varepsilon}$ for every $\varepsilon > 0$.

[Dreier, Kuinke, Rossmanith 2020]
A random graph model is \textit{3-power-law-bounded} if (roughly speaking):

- fraction of vertices with degree $k$ is $O(k^{-3})$

real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$

[Dreier, Kuinke, Rossmanith 2020]
Our Result

A random graph model is 3-	extit{power-law-bounded} if (roughly speaking):

- fraction of vertices with degree $k$ is $O(k^{-3})$
  
  real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$

- unclustered
  
  real networks: typically clustered

[Dreier, Kuinke, Rossmanith 2020]
Our Result

A random graph model is \textit{3-power-law-bounded} if (roughly speaking):

- fraction of vertices with degree $k$ is $O(k^{-3})$
  
  \textbf{real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$}

- unclustered
  
  \textbf{real networks: typically clustered}

\textbf{Theorem}

\textit{Given a first-order sentence $\varphi$ and a graph $G$ sampled from a 3-power-law-bounded model, one can decide whether $\varphi$ is true on $G$ in expected time $f(|\varphi|)n^{1+\varepsilon}$ for every $\varepsilon > 0$.}

[Dreier, Kuinke, Rossmanith 2020]
A more direct way to get a desirable degree distribution.

\[ \frac{1}{\sqrt{i \cdot j}} \]

expected degree \( \approx \sqrt{\frac{n}{i}} \)
A random graph model with vertices $1, \ldots, n$ is \textit{$\alpha$-power-law-bounded} if the probability that some subset of edges $E \subseteq \left(\{1, \ldots, n\}\right)$ is present is at most

$$\log(n) O\left(|E|^\alpha\right) \prod_{ij \in E} \frac{1}{\sqrt{i \cdot j}}.$$
A random graph model with vertices 1, \ldots, n is \textit{3-power-law-bounded} if the probability that some subset of edges $E \subseteq \binom{\{1,\ldots,n\}}{2}$ is present is at most

$$\prod_{ij \in E} \frac{1}{\sqrt{i \cdot j}}.$$
A random graph model with vertices 1, \ldots, n is \textit{3-power-law-bounded} if the probability that some subset of edges $E \subseteq \binom{\{1,\ldots,n\}}{2}$ is present is at most

$$\log(n)^{O(|E|^2)} \prod_{ij \in E} \frac{1}{\sqrt{i \cdot j}}.$$
A random graph model with vertices 1, \ldots, n is \textit{3-power-law-bounded} if the probability that some subset of edges $E \subseteq \binom{\{1, \ldots, n\}}{2}$ is present is at most

$$\log(n)^O(|E|^2) \prod_{i,j \in E} \frac{1}{\sqrt{i \cdot j}}.$$
Asymptotic Structure of 3-power-law-bounded models
Asymptotic Structure of 3-power-law-bounded models

radius: $r$
Asymptotic Structure of 3-power-law-bounded models

\[ \log(n)O(r^2) \]
Asymptotic Structure of 3-power-law-bounded models

\[ \log(n)O(r^2) \]

radius: \( r \)
Asymptotic Structure of 3-power-law-bounded models

The diagram represents the relationship between various elements, with the central node labeled \( \log(n)O(r^2) \). The labels around the central node indicate the order of magnitude:

- \( O(r) \)
- \( O(1) \)
- \( O(r) \)
- \( O(r) \)
- \( O(r) \)
- \( O(r) \)

The diagram also includes an annotation indicating the radius: \( r \).
Asymptotic Structure of 3-power-law-bounded models

\[ \log(n)O(r^2) \]
Algorithm

Input: graph sampled from 3-power-law-bounded model
Algorithm
Gaifman’s theorem: consider only neighborhoods
Gaifman’s theorem: consider only neighborhoods
approximately find core
prune trees
prune protrusions
use brute force on core
repeat for every neighborhood
Done!
Gaifman’s theorem: consider only neighborhoods
our run time:
our run time:

\[
2^{\overbrace{222222}^{\varphi}} \cdot n^{1+\varepsilon}
\]
○ our run time:

\[
\underbrace{2^{2^22^22^22^22^22^2}}_{|\varphi|} \cdot n^{1+\varepsilon}
\]

○ under worst-case complexity this is optimal
- our run time:

\[ \underbrace{2 \times 2 \times 2 \times 2 \times 2 \times 2} \cdot n^{1+\varepsilon} \]

- under worst-case complexity this is optimal

- open question: can we do better in the average-case?
A random graph model is 3-power-law-bounded if (roughly speaking):
A random graph model is \textit{3-power-law-bounded} if (roughly speaking):

- fraction of vertices with degree \( k \) is \( O(k^{-\alpha}) \)
- real networks: typically \( k^{-\alpha} \) with \( 2 \leq \alpha \leq 3 \)
A random graph model is \textit{3-power-law-bounded} if (roughly speaking):

- fraction of vertices with degree $k$ is $O(k^{-3})$
  
  \textbf{real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$}

- unclustered
  
  \textbf{real networks: typically clustered}
Summary (so far)

A random graph model is *3-power-law-bounded* if (roughly speaking):

- fraction of vertices with degree $k$ is $O(k^{-3})$
  - real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$

- unclustered
  - real networks: typically clustered

Theorem

*Given a first-order sentence $\varphi$ and a graph $G$ sampled from a 3-power-law-bounded model, one can decide whether $\varphi$ is true on $G$ in expected time $f(|\varphi|)n^{1+\varepsilon}$ for every $\varepsilon > 0$.***
A random graph model is 3-power-law-bounded if (roughly speaking):

- fraction of vertices with degree \( k \) is \( O(k^{-3}) \)
- real networks: typically \( k^{-\alpha} \) with \( 2 \leq \alpha \leq 3 \)

- unclustered
  - real networks: typically clustered

**Theorem**

Given a first-order sentence \( \varphi \) and a graph \( G \) sampled from a 3-power-law-bounded model, one can decide whether \( \varphi \) is true on \( G \) in expected time \( f(|\varphi|)n^{1+\varepsilon} \) for every \( \varepsilon > 0 \).
Summary (so far)

A random graph model is \textit{3-power-law-bounded} if (roughly speaking):

- fraction of vertices with degree \( k \) is \( O(k^{-3}) \)
  - real networks: typically \( k^{-\alpha} \) with \( 2 \leq \alpha \leq 3 \)

- unclustered
  - real networks: typically clustered

Theorem

\textit{Given a first-order sentence} \( \varphi \) \textit{and a graph} \( G \) \textit{sampled from a 3-power-law-bounded model, one can decide whether} \( \varphi \) \textit{is true on} \( G \) \textit{in expected time} \( f(|\varphi|)n^{1+\epsilon} \) \textit{for every} \( \epsilon > 0 \).
A random graph model is \emph{3-power-law-bounded} if
(roughly speaking):

- fraction of vertices with degree $k$ is $O(k^{-3})$
  real networks: typically $k^{-\alpha}$ with $2 \leq \alpha \leq 3$

- unclustered
  real networks: typically clustered

**Theorem**

\textit{Given a first-order sentence $\varphi$ and a graph $G$ sampled from a 3-power-law-bounded model, one can decide whether $\varphi$ is true on $G$ in expected time $f(|\varphi|)n^{1+\varepsilon}$ for every $\varepsilon > 0$.}

\textit{Big Question:} model-checking on clustered models?
Want a Challenge?

Can we solve the problem on *Hyperbolic Random Graphs*?
LOWER BOUNDS
Average Case Complexity

- worst-case: P and NP
Average Case Complexity

- worst-case: P and NP
- average-case: avgP and (NP, PComp)?
Average Case Complexity

- worst-case: P and NP
- average-case: avgP and (NP, PComp)?
- expected time example:

\[ \sum_{x \in \mathbb{N}} P[X = x] \cdot \text{time}(x) = O(1) \]
With Colors
With Colors
With Colors

\[ \forall x \exists y \ldots \]

\[ E[\text{time}] \leq f(|\varphi|)n^{1+\varepsilon} \]
$\forall x \exists y \ldots$

$E[\text{time}] \leq f(|\varphi|)n^{1+\varepsilon}$?
What if $\alpha < 3$?

[Dreier, Rossmanith 2019]
Without Colors
Without Colors

∀x ∃y ...
Zero-One Laws

1/2 random graph
Zero-One Laws

\[ \Pr[ \text{1/2 random graph} \in \text{graph} ] \rightarrow 1 \]
Zero-One Laws

\[ \Pr[ \text{triangle} \in \text{graph} ] \to 1 \]

\[ \Pr[ \text{apex vertex} \in \text{graph} ] \to 0 \]
Zero-One Laws

\[ \Pr[ \begin{array}{c} \text{apex} \\ \text{vertex} \end{array} \in \begin{array}{c} \text{ } \\ \text{ } \end{array} ] \rightarrow 0 \]

\[ \Pr[ \begin{array}{c} \triangle \in \begin{array}{c} \text{ } \\ \text{ } \end{array} \ ] \rightarrow 1 \]

for all \( \varphi \) \( \Pr[ \begin{array}{c} \text{ } \\ \text{ } \end{array} \models \varphi ] \rightarrow \{0, 1\} \)
Average-Case Hierarchy Collapse

- worst-case
  - W[1]-complete:

\[ \exists x_1 \ldots \exists x_k \]

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - $W[2]$-complete:

$$\exists x_1 \ldots \exists x_k \forall y$$

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - $W[3]$-complete:
    \[ \exists x_1 \ldots \exists x_k \forall y \exists z \]

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - $W[4]$-complete:
    $$\exists x_1 \ldots \exists x_k \forall y \exists z \forall w$$

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - $W[4]$-complete:
    $$\exists x_1 \ldots \exists x_k \forall y \exists z \forall w$$

- average-case on 1/2 random graph

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - \( W[4]\)-complete:
    \[
    \exists x_1 \ldots \exists x_k \forall y \exists z \forall w
    \]

- average-case on 1/2 random graph
  - FPT (i.e., \( f(|\varphi|)n \) time):
    \[
    \exists x_1 \ldots \exists x_k
    \]

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - $W[4]$-complete:
    \[
    \exists x_1 \ldots \exists x_k \forall y \exists z \forall w
    \]

- average-case on 1/2 random graph
  - FPT (i.e., $f(|\varphi|)n$ time):
    \[
    \exists x_1 \ldots \exists x_k
    \]
  - equally hard:
    \[
    \exists x_1 \ldots \exists x_k \forall y
    \]

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - $W[4]$-complete:
    $$\exists x_1 \ldots \exists x_k \forall y \exists z \forall w$$

- average-case on 1/2 random graph
  - FPT (i.e., $f(|\varphi|)n$ time):
    $$\exists x_1 \ldots \exists x_k$$
  - equally hard:
    $$\exists x_1 \ldots \exists x_k \forall y \exists z$$

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - \( W[4]\)-complete:
    \[
    \exists x_1 \ldots \exists x_k \forall y \exists z \forall w
    \]

- average-case on 1/2 random graph
  - FPT (i.e., \( f(|\varphi|)n \) time):
    \[
    \exists x_1 \ldots \exists x_k
    \]
  - equally hard:
    \[
    \exists x_1 \ldots \exists x_k \forall y \exists z \forall w
    \]

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

○ worst-case
  ○ $W[4]$-complete:
    \[
    \exists x_1 \ldots \exists x_k \forall y \exists z \forall w
    \]

○ average-case on 1/2 random graph
  ○ FPT (i.e., $f(|\varphi|)n$ time):
    \[
    \exists x_1 \ldots \exists x_k
    \]
  ○ equally hard:
    \[
    \exists x_1 \ldots \exists x_k \forall y \exists z \forall w \exists u
    \]

[Dreier, Lotze, Rossmanith 2020]
Average-Case Hierarchy Collapse

- worst-case
  - W[4]-complete:
    \[ \exists x_1 \ldots \exists x_k \forall y \exists z \forall w \]

- average-case on 1/2 random graph
  - FPT (i.e., \( f(|\varphi|)n \) time):
    \[ \exists x_1 \ldots \exists x_k \]
  - equally hard:
    \[ \exists x_1 \ldots \exists x_k \forall y \exists z \forall w \exists u \ldots \]

[Dreier, Lotze, Rossmanith 2020]
Solve all problems definable in FO logic on unclustered graphs that are not too skewed.
Solve all problems definable in FO logic on unclustered graphs that are not too skewed.

How about clustered models?
Summary

- Solve all problems definable in FO logic on unclustered graphs that are not too skewed.
- How about clustered models?
- How about (uncolored) lower bounds?
Thanks!

dreier@ac.tuwien.ac.at