HUMBOLDT-UNIVERSITÄT ZU BERLIN

Algorithmic Metatheorems for Clique-Guarded First-Order Logic with Counting

Steffen van Bergerem, Johannes Lange, and Nicole Schweikardt

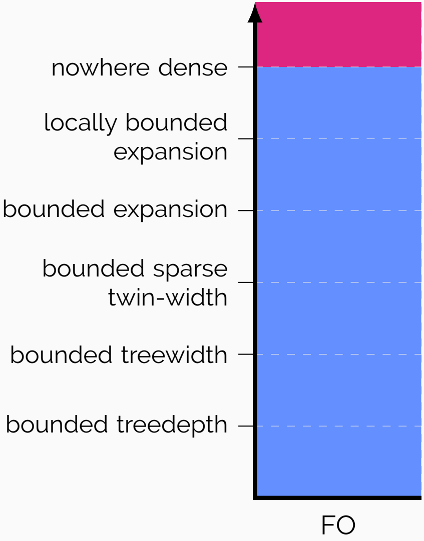
AlMoTh 2026

First-Order Model Checking

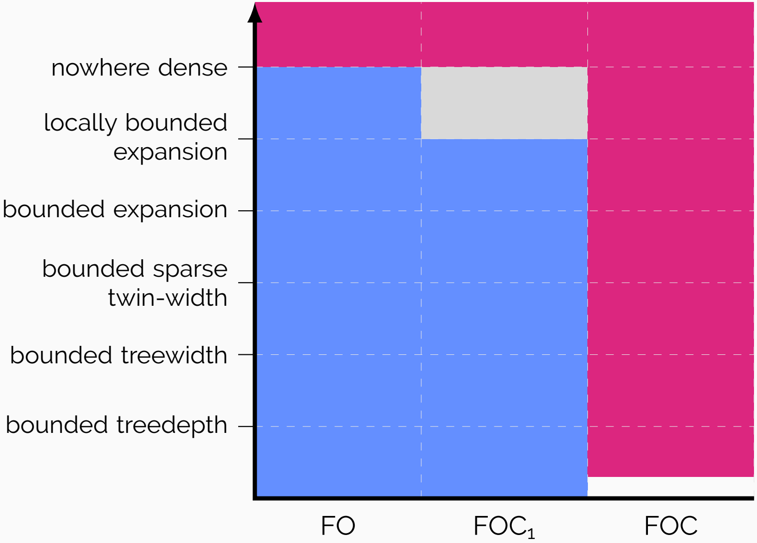
Given a relational structure $\mathcal{A}$ and an FO sentence φ

Decide whether $\mathcal{A} \models \varphi$

Model Checking on Sparse Classes

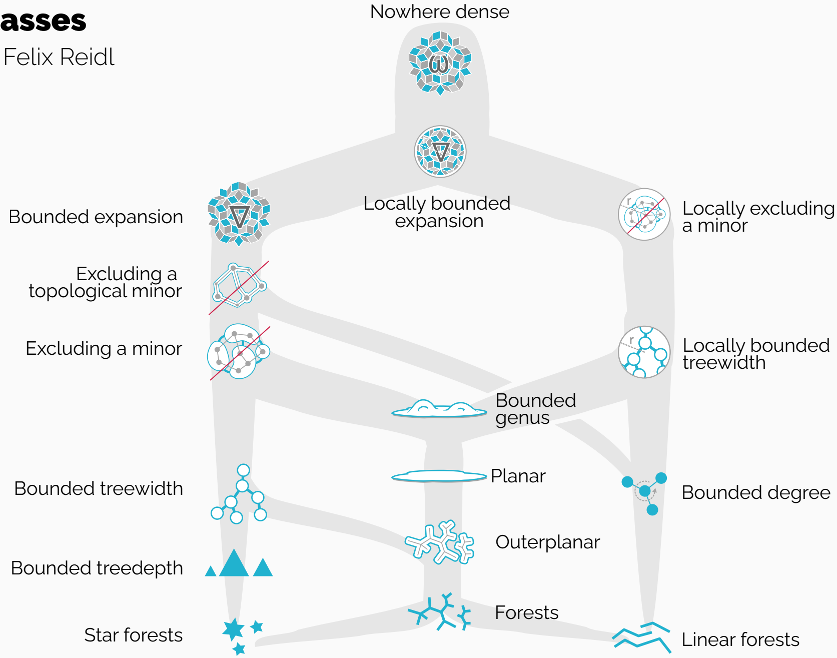


Model Checking on Sparse Classes



Sparse Classes

Illustration by Felix Reidl



First-Order Logic with Counting (FOC)

Counting terms

- i for every integer $i \in \mathbb{Z}$
- $\#(y_1, \ldots, y_k).\varphi(\bar{x}, \bar{y})$ for every FOC formula $\varphi(\bar{x}, \bar{y})$
- $t_1 + t_2$ and $t_1 \cdot t_2$ for all FOC counting terms t_1, t_2

Formulae

First-Order Logic with Counting (FOC)

Counting terms

- i for every integer $i \in \mathbb{Z}$
- $\#(y_1, \dots, y_k).\varphi(\bar{x}, \bar{y})$ for every FOC formula $\varphi(\bar{x}, \bar{y})$
- $t_1 + t_2$ and $t_1 \cdot t_2$ for all FOC counting terms t_1, t_2

Formulae

- every FO formula
- $\neg\varphi_1$, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $\exists x \varphi_1$ for all FOC formulae φ_1, φ_2
- $P(t_1, \dots, t_m)$ for all FOC counting terms $t_1, \dots, t_m$ and $P \in \mathbb{P}$, $P \subseteq \mathbb{Z}^m$

First-Order Logic with Counting (FOC)

Counting terms

- $t_1(x) = \#(y).E(x,y)$
- $t_2 = \#(x_1, \ldots, x_k).(\bigwedge_{1 \leqslant i < j \leqslant k} E(x_i, x_j))$

Formulae

First-Order Logic with Counting (FOC)

Counting terms

- $t_1(x) = \#(y).E(x,y)$
- $t_2 = \#(x_1,\ldots,x_k).(\bigwedge_{1\leqslant i<j\leqslant k} E(x_i,x_j))$

Formulae

- $\varphi_1(x) = (\#(y).E(x,y) \leqslant \#(y).\neg E(x,y))$
- $\varphi_2(x) = \exists y \left(\#(z).(E(x,z) \wedge E(z,y)) \geqslant 2\right)$

First-Order Logic with Counting (FOC)

Counting terms

- $t_1(x) = \#(y).E(x,y)$
- $t_2 = \#(x_1,\ldots,x_k).(\bigwedge_{1\leqslant i<j\leqslant k} E(x_i,x_j))$
- $t_3 = t_2 + \#(x).\varphi_1(x)$

Formulae

- $\varphi_1(x) = (\#(y).E(x,y) \leqslant \#(y).\neg E(x,y))$
- $\varphi_2(x) = \exists y \left(\#(z).(E(x,z) \wedge E(z,y)) \geqslant 2\right)$

First-Order Logic with Counting (FOC)

Counting terms

- $t_1(x) = \#(y).E(x,y)$
- $t_2 = \#(x_1, \dots, x_k).(\bigwedge_{1 \leqslant i < j \leqslant k} E(x_i, x_j))$
- $t_3 = t_2 + \#(x).\varphi_1(x)$

Formulae

- $\varphi_1(x) = (\#(y).E(x,y) \leqslant \#(y).\neg E(x,y))$
- $\varphi_2(x) = \exists y \left(\#(z).(E(x,z) \wedge E(z,y)) \geqslant 2\right)$
- $\varphi_3(x) = \exists y\, E(x,y) \wedge (t_1(x) \leqslant t_1(y))$

First-Order Logic with Counting (FOC)

Counting terms

- $t_1(x) = \#(y).E(x,y)$
- $t_2 = \#(x_1,\dots,x_k).(\bigwedge_{1\leqslant i<j\leqslant k} E(x_i,x_j))$
- $t_3 = t_2 + \#(x).\varphi_1(x)$

Formulae

- $\varphi_1(x) = (\#(y).E(x,y) \leqslant \#(y).\neg E(x,y))$
- $\varphi_2(x) = \exists y\left(\#(z).(E(x,z) \wedge E(z,y)) \geqslant 2\right)$
- $\varphi_3(x) = \exists y\, E(x,y) \wedge (t_1(x) \leqslant t_1(y))$

FOC$_1$

- introduced by Grohe and Schweikardt (PODS 2018)
- subformulae comparing counting terms have $\leqslant 1$ free variable

First-Order Logic with Counting (FOC)

Counting terms

- $t_1(x) = \#(y).E(x,y)$
- $t_2 = \#(x_1,\ldots,x_k).(\bigwedge_{1\leqslant i<j\leqslant k} E(x_i,x_j))$
- $t_3 = t_2 + \#(x).\varphi_1(x)$

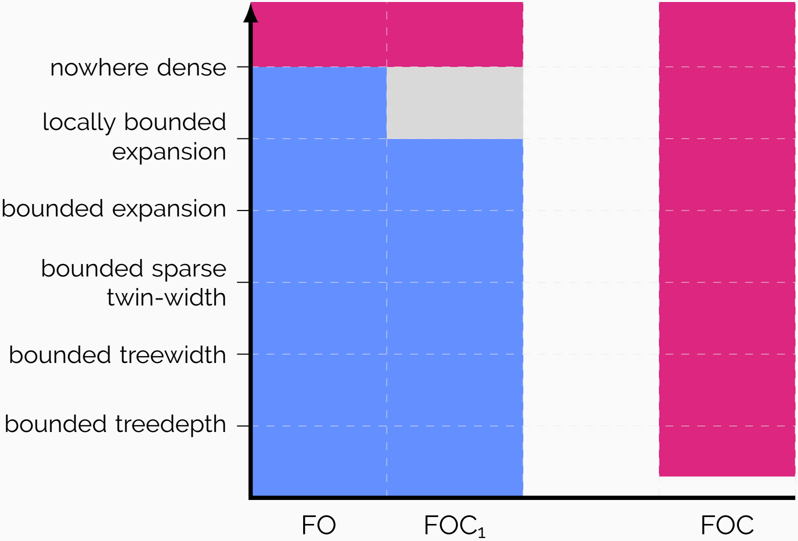
Formulae

- $\varphi_1(x) = (\#(y).E(x,y) \leqslant \#(y).\neg E(x,y))$
- $\varphi_2(x) = \exists y \left(\#(z).(E(x,z) \wedge E(z,y)) \geqslant 2\right)$
- $\varphi_3(x) = \exists y\, E(x,y) \wedge (t_1(x) \leqslant t_1(y))$

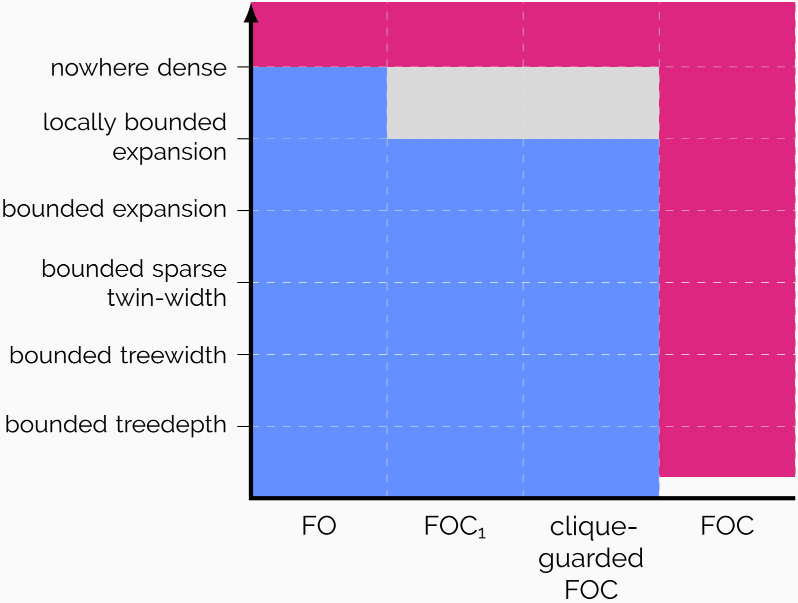
FOC_1

- introduced by Grohe and Schweikardt (PODS 2018)
- subformulae comparing counting terms have $\leqslant 1$ free variable

Evaluation on Sparse Classes



Evaluation on Sparse Classes



Clique-Guarded First-Order Logic with Counting (cgFOC)

FOC_1
subformulae comparing counting terms have $\leqslant 1$ free variable

Clique-Guarded First-Order Logic with Counting (cgFOC)

FOC$_1$
subformulae comparing counting terms have $\leqslant 1$ free variable

clique-guarded FOC (“free variables have distance at most 1”)
every pair of free variables in subformulae comparing counting terms is guarded by an atom

Clique-Guarded First-Order Logic with Counting (cgFOC)

FOC$_1$
subformulae comparing counting terms have $\leqslant$ 1 free variable

clique-guarded FOC (“free variables have distance at most 1”)
every pair of free variables in subformulae comparing counting terms is guarded by an atom

- $\varphi_1(x_1, x_2, x_3) = E(x_1, x_2) \wedge E(x_1, x_3) \wedge E(x_2, x_3) \wedge \big(t_1(x_1, x_2) = t_2(x_3)\big)$
  for cgFOC counting terms t_1, t_2

Clique-Guarded First-Order Logic with Counting (cgFOC)

FOC$_1$
subformulae comparing counting terms have $\leqslant 1$ free variable

clique-guarded FOC ("free variables have distance at most 1")
every pair of free variables in subformulae comparing counting terms is guarded by an atom

- $\varphi_1(x_1, x_2, x_3) = E(x_1, x_2) \wedge E(x_1, x_3) \wedge E(x_2, x_3) \wedge (t_1(x_1, x_2) = t_2(x_3))$
  for cgFOC counting terms t_1, t_2
- $\varphi_2(x_1, x_2, x_3, x_4) = R(x_1, x_3, x_4) \wedge R(x_2, x_3, x_4) \wedge E(x_1, x_2)$
  $\wedge\ (t_1(x_1, x_4) + t_2(x_2) = t_1(x_2, x_4))$

Clique-Guarded First-Order Logic with Counting (cgFOC)

FOC$_1$
subformulae comparing counting terms have $\leqslant 1$ free variable

clique-guarded FOC ("free variables have distance at most 1")
every pair of free variables in subformulae comparing counting terms is guarded by an atom

- $\varphi_1(x_1, x_2, x_3) = E(x_1, x_2) \wedge E(x_1, x_3) \wedge E(x_2, x_3) \wedge (t_1(x_1, x_2) = t_2(x_3))$
  for cgFOC counting terms t_1, t_2
- $\varphi_2(x_1, x_2, x_3, x_4) = R(x_1, x_3, x_4) \wedge R(x_2, x_3, x_4) \wedge E(x_1, x_2)$
  $\wedge\ (t_1(x_1, x_4) + t_2(x_2) = t_1(x_2, x_4))$
- $t(\bar{x}) = \#\bar{y}.\varphi(\bar{x}, \bar{y})$ for a cgFOC formula φ

Clique-Guarded First-Order Logic with Counting (cgFOC)

FOC$_1$
subformulae comparing counting terms have $\leqslant 1$ free variable

clique-guarded FOC (“free variables have distance at most 1”)
every pair of free variables in subformulae comparing counting terms is guarded by an atom

- $\varphi_1(x_1, x_2, x_3) = E(x_1, x_2) \wedge E(x_1, x_3) \wedge E(x_2, x_3) \wedge (t_1(x_1, x_2) = t_2(x_3))$
  for cgFOC counting terms t_1, t_2
- $\varphi_2(x_1, x_2, x_3, x_4) = R(x_1, x_3, x_4) \wedge R(x_2, x_3, x_4) \wedge E(x_1, x_2)$
  $\wedge\ (t_1(x_1, x_4) + t_2(x_2) = t_1(x_2, x_4))$
- $t(\bar{x}) = \#\bar{y}.\varphi(\bar{x}, \bar{y})$ for a cgFOC formula φ

not clique-guarded FOC

- $\psi(x_1, x_2, x_3) = E(x_1, x_2) \wedge E(x_2, x_3) \wedge (t(x_1) \leqslant t(x_3))$

Main Results

Main Results

Tractability

- query answering
- enumeration
- agnostic PAC learning

Hardness

- model checking for variations of cgFOC

Query Answering

Preprocessing

Given a σ-structure $\mathcal{A}$,
a cgFOC expression $\xi(\bar{x})$

Query answering

Given a tuple $\bar{v} \in A^{|\bar{x}|}$

Output $[\![\xi(\bar{v})]\!]^{\mathcal{A}}$

Enumeration

Preprocessing

Given a σ-structure $\mathcal{A}$,
a cgFOC formula $\xi(\bar{x})$

Enumeration

Output all tuples $\bar{v} \in A^{|\bar{x}|}$ with $\mathcal{A} \models \xi(\bar{v})$ in lexicographic order and without duplicates

Query Answering

Preprocessing

Given a σ-structure $\mathcal{A}$,
a cgFOC expression $\xi(\bar{x})$

Query answering

Given a tuple $\bar{v} \in A^{|\bar{x}|}$

Output $[\![\xi(\bar{v})]\!]^{\mathcal{A}}$

Enumeration

Preprocessing

Given a σ-structure $\mathcal{A}$,
a cgFOC formula $\xi(\bar{x})$

Enumeration

Output all tuples $\bar{v} \in A^{|\bar{x}|}$ with $\mathcal{A} \models \xi(\bar{v})$ in lexicographic order and without duplicates

v. B., Lange, and Schweikardt, 2026+

For every class $\mathcal{C}$ of **locally bounded expansion**, there are algorithms that solve the **query-answering** || **enumeration** problem for cgFOC on $\mathcal{C}$ with

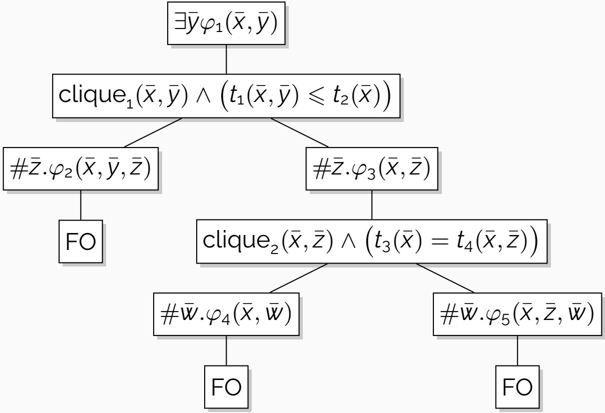
- preprocessing in time $f(\xi, \sigma, \varepsilon) \cdot |A|^{1+\varepsilon}$
- query answering in time $f(\xi, \sigma, \varepsilon)$ || $f(\xi, \sigma, \varepsilon)$ delay.

Query Answering and Enumeration: Main Proof Ideas

- Toruńczyk (PODS 2020): query answering on bounded expansion for queries of the form $\#\bar{y}.\varphi(\bar{x},\bar{y})$ for an FO formula φ
- we generalise this to classes of locally bounded expansion

Query Answering and Enumeration: Main Proof Ideas

- Toruńczyk (PODS 2020): query answering on bounded expansion for queries of the form $\#\bar{y}.\varphi(\bar{x},\bar{y})$ for an FO formula φ
- we generalise this to classes of locally bounded expansion



$\exists\bar{y}\varphi_1(\bar{x},\bar{y})$

$\mathrm{clique}_1(\bar{x},\bar{y}) \wedge \big(t_1(\bar{x},\bar{y}) \leqslant t_2(\bar{x})\big)$

$\#\bar{z}.\varphi_2(\bar{x},\bar{y},\bar{z})$

$\#\bar{z}.\varphi_3(\bar{x},\bar{z})$

FO

$\mathrm{clique}_2(\bar{x},\bar{z}) \wedge \big(t_3(\bar{x}) = t_4(\bar{x},\bar{z})\big)$

$\#\bar{w}.\varphi_4(\bar{x},\bar{w})$

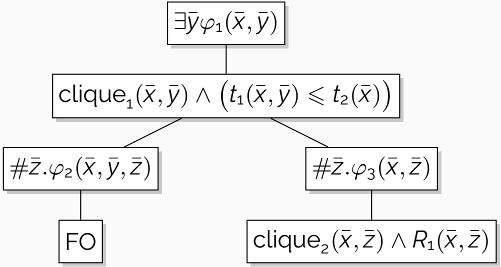
$\#\bar{w}.\varphi_5(\bar{x},\bar{z},\bar{w})$

FO

FO

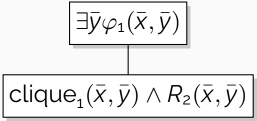
Query Answering and Enumeration: Main Proof Ideas

- Toruńczyk (PODS 2020): query answering on bounded expansion for queries of the form $\#\bar{y}.\varphi(\bar{x}, \bar{y})$ for an FO formula φ
- we generalise this to classes of locally bounded expansion


$\exists \bar{y} \varphi_1(\bar{x}, \bar{y})$
$\text{clique}_1(\bar{x}, \bar{y}) \wedge \big(t_1(\bar{x}, \bar{y}) \leqslant t_2(\bar{x})\big)$
$\#\bar{z}.\varphi_2(\bar{x}, \bar{y}, \bar{z})$
$\#\bar{z}.\varphi_3(\bar{x}, \bar{z})$
FO
$\text{clique}_2(\bar{x}, \bar{z}) \wedge R_1(\bar{x}, \bar{z})$

Query Answering and Enumeration: Main Proof Ideas

- Toruńczyk (PODS 2020): query answering on bounded expansion for queries of the form $\#\bar{y}.\varphi(\bar{x},\bar{y})$ for an FO formula φ
- we generalise this to classes of locally bounded expansion

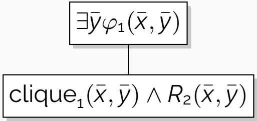


$\exists\bar{y}\varphi_1(\bar{x},\bar{y})$

$\text{clique}_1(\bar{x},\bar{y}) \wedge R_2(\bar{x},\bar{y})$

Query Answering and Enumeration: Main Proof Ideas

- Toruńczyk (PODS 2020): query answering on bounded expansion for queries of the form $\#\bar{y}.\varphi(\bar{x},\bar{y})$ for an FO formula φ
- we generalise this to classes of locally bounded expansion



$\exists\bar{y}\varphi_1(\bar{x},\bar{y})$

$\text{clique}_1(\bar{x},\bar{y}) \wedge R_2(\bar{x},\bar{y})$

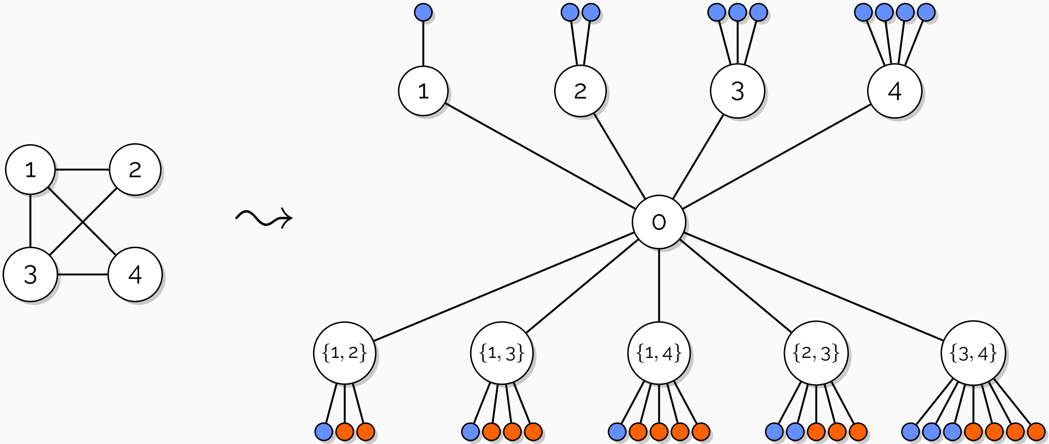
- iteratively replace subformulae comparing counting terms by new relation symbols
- because of "clique-guarded", when computing the new relations,
  - we only need to consider tuples that form a clique,
    number of those is small in classes of locally bounded expansion
  - the new relations do not change the Gaifman graph
- in the end, we only need to handle a first-order formula

Hardness

v. B., Lange, and Schweikardt, 2026+

If we allow constructs of the form $E(x_1, x_2) \wedge E(x_2, x_3) \wedge (t(x_1) = t'(x_3))$, then the model-checking problem for the resulting logic (**“2-guarded” FOC**) is $\mathrm{AW}[*]$-**hard** on the class of coloured **trees of height 2**.

Hardness of 2-Guarded FOC



Hardness

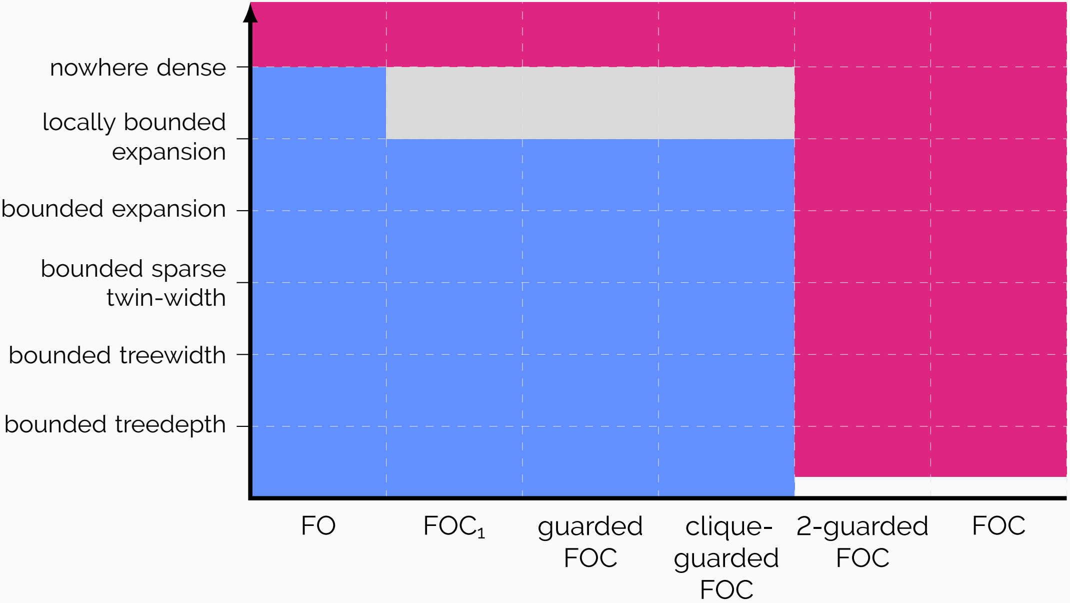
v. B., Lange, and Schweikardt, 2026+

If we allow constructs of the form $E(x_1, x_2) \wedge E(x_2, x_3) \wedge (t(x_1) = t'(x_3))$, then the model-checking problem for the resulting logic (**"2-guarded" FOC**) is AW$[*]$-**hard** on the class of coloured **trees of height 2**.

v. B., Lange, and Schweikardt, 2026+

Even if we only allow constructs of the form $E(x_1, x_2) \wedge (t(x_1) = t'(x_2))$, the model-checking problem for the resulting logic (**"guarded" FOC**) is AW$[*]$-**hard** on a class of coloured **graphs of shrub-depth 2**.

Evaluation on Sparse Classes



Agnostic PAC Learning

- agnostic Probably Approximately Correct learning
- fix $k \in \mathbb{N}_{\geqslant 1}, \ell, q \in \mathbb{N}$
- assume probability distribution $\mathcal{D}$ on $A^k \times \{0, 1\}$
- we want to approximate $\mathcal{D}$ by a cgFOC formula

Agnostic PAC Learning

- agnostic Probably Approximately Correct learning
- fix $k \in \mathbb{N}_{\geqslant 1}$, $\ell, q \in \mathbb{N}$
- assume probability distribution $\mathcal{D}$ on $A^k \times \{0, 1\}$
- we want to approximate $\mathcal{D}$ by a cgFOC formula
- algorithm is given a structure $\mathcal{A}$ and draws examples from the distribution
- **Goal:** return a cgFOC formula $\varphi(x_1, \ldots, x_k)$ with $\mathrm{qr}(\varphi) \leqslant q$ that uses at most ℓ vertices from A as constants and has small expected error

$$\Pr_{(\bar{v},\lambda)\sim\mathcal{D}}\left(\llbracket\varphi(\bar{v})\rrbracket^{\mathcal{A}} \neq \lambda\right)$$

Agnostic PAC Learning

v. B., Lange, and Schweikardt, 2026+

On every class of locally bounded expansion, there is an algorithm that solves the agnostic-PAC-learning problem for cgFOC in time $\mathcal{O}(|A|^{1+\varepsilon})$.

Agnostic PAC Learning

v. B., Lange, and Schweikardt, 2026+

On every class of locally bounded expansion, there is an algorithm that solves the agnostic-PAC-learning problem for cgFOC in time $\mathcal{O}(|A|^{1+\varepsilon})$.

- we even solve a more powerful problem, which we call **agnostic PAC enumeration**, with $\mathcal{O}(|A|^{1+\varepsilon})$ preprocessing and $\mathcal{O}(1)$ delay
- this enumerates all formulae within the given complexity bounds, sorted by their (approximate) error

Agnostic PAC Learning

v. B., Lange, and Schweikardt, 2026+

On every class of locally bounded expansion, there is an algorithm that solves the agnostic-PAC-learning problem for cgFOC in time $\mathcal{O}(|A|^{1+\varepsilon})$.

- we even solve a more powerful problem, which we call **agnostic PAC enumeration**, with $\mathcal{O}(|A|^{1+\varepsilon})$ preprocessing and $\mathcal{O}(1)$ delay
- this enumerates all formulae within the given complexity bounds, sorted by their (approximate) error

Proof idea

- cgFOC formulae have bounded VC dimension on classes of locally bounded expansion (next talk)
- implies that constant number of examples suffices for approximation
- encode these examples in the structure, use enumeration result

Conclusion

Tractability for cgFOC on classes of locally bounded expansion

- $f(\xi, \sigma, \varepsilon)$-time **query answering** after $f(\xi, \sigma, \varepsilon) \cdot |A|^{1+\varepsilon}$-time preprocessing
- $f(\xi, \sigma, \varepsilon)$-delay **enumeration** after $f(\xi, \sigma, \varepsilon) \cdot |A|^{1+\varepsilon}$-time preprocessing
- $\mathcal{O}(1)$-delay **agnostic PAC enumeration** after $\mathcal{O}(|A|^{1+\varepsilon})$-time preprocessing

Hardness

- "2-guarded" FOC model checking is AW$[*]$-hard on trees of height 2
- "guarded" FOC model checking is AW$[*]$-hard on a class of shrub-depth 2

Conclusion

Tractability for cgFOC on classes of locally bounded expansion

- $f(\xi,\sigma,\varepsilon)$-time **query answering** after $f(\xi,\sigma,\varepsilon)\cdot|A|^{1+\varepsilon}$-time preprocessing
- $f(\xi,\sigma,\varepsilon)$-delay **enumeration** after $f(\xi,\sigma,\varepsilon)\cdot|A|^{1+\varepsilon}$-time preprocessing
- $\mathcal{O}(1)$-delay **agnostic PAC enumeration** after $\mathcal{O}(|A|^{1+\varepsilon})$-time preprocessing

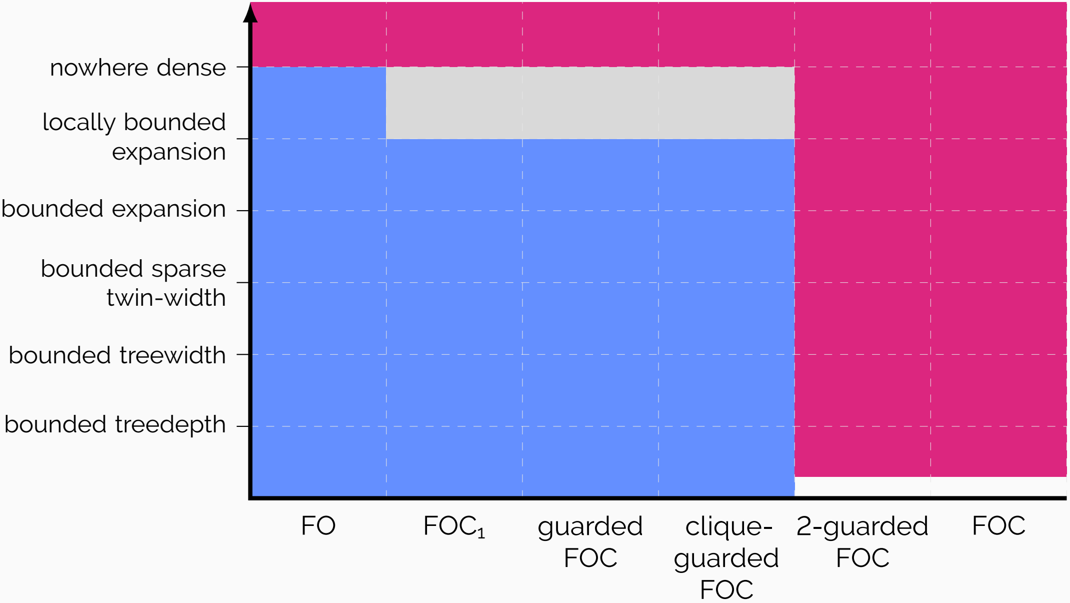
Hardness

- "2-guarded" FOC model checking is AW$[*]$-hard on trees of height 2
- "guarded" FOC model checking is AW$[*]$-hard on a class of shrub-depth 2

Open Questions

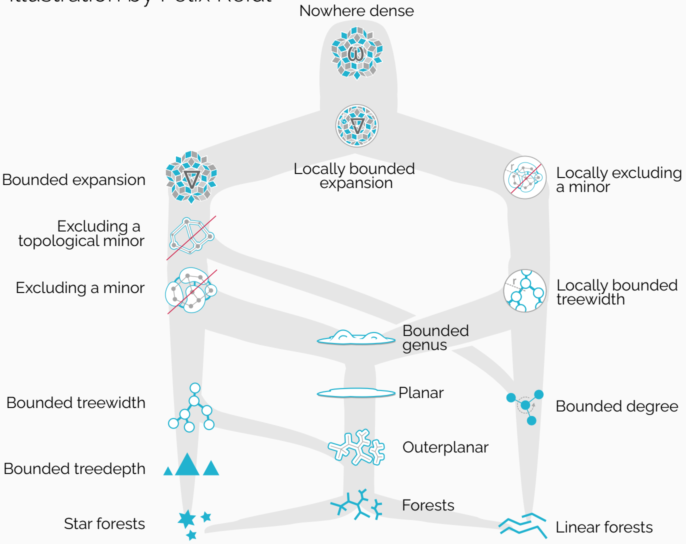
- Tractability on nowhere dense classes?
- FOC_1 on dense classes?

Evaluation on Sparse Classes



Sparse Classes

Illustration by Felix Reidl



depth-r minor
contract subgraphs of radius at most r

bounded expansion
for every $G \in \mathcal{C}$, $r \in \mathbb{N}$, and every depth-r minor H of G, $\frac{|E(H)|}{|V(H)|} \leqslant g(r)$

locally bounded expansion
for every $d \in \mathbb{N}$, the class of neighbourhoods of radius d in graphs from $\mathcal{C}$ has bounded expansion

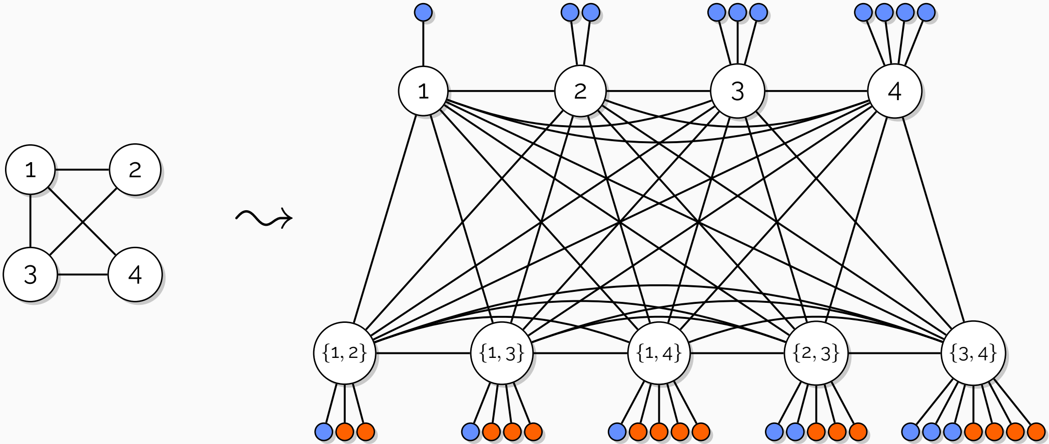
Number of Cliques in Sparse Classes

Lemma

Let $\mathcal{C}$ be a nowhere dense graph class. There is a function f such that for every $k \in \mathbb{N}_{\geqslant 1}$, $\varepsilon \in \mathbb{Q}_{>0}$, and $G \in \mathcal{C}$, the number of k-cliques in G is at most $f(k, \varepsilon) \cdot |V(G)|^{1+\varepsilon}$.

- every graph in $\mathcal{C}$ is $\left(f'(\varepsilon) \cdot |V(G)|^{\varepsilon}\right)$-degenerate
- d-degenerate graphs contain at most $d^k \cdot |V(G)|$ many cliques of size k

Hardness of Guarded FOC


1
2
3
4
⇝
1
2
3
4
$\{1,2\}$
$\{1,3\}$
$\{1,4\}$
$\{2,3\}$
$\{3,4\}$