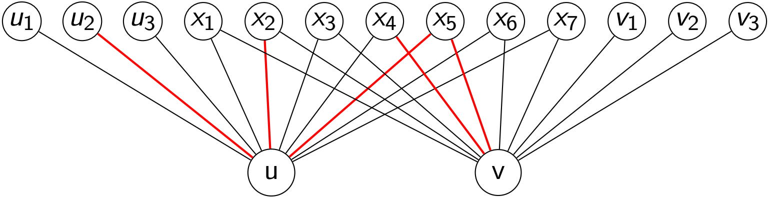
Twin-width I: tractable FO model checking

Édouard Bonnet, Eun Jung Kim,
Stéphan Thomassé, and Rémi Watrigant

ENS Lyon, LIP

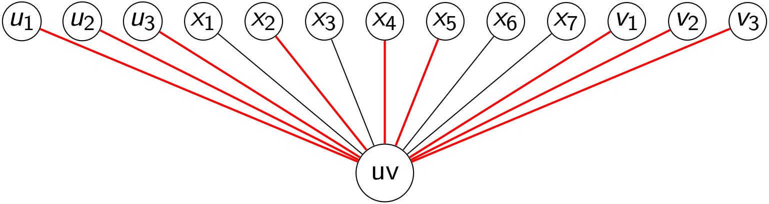
FOCS 2020

Trigraph and contractions



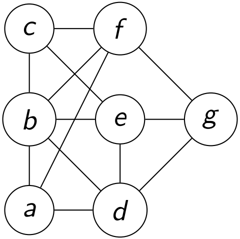
Trigraph: non-edges, edges, and red edges (error)

Trigraph and contractions



edges to $N(u) \triangle N(v)$ turn red, for $N(u) \cap N(v)$ red is absorbing

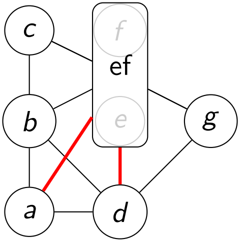
Contraction sequence and twin-width



Maximum red degree $= 0$

overall maximum red degree = 0

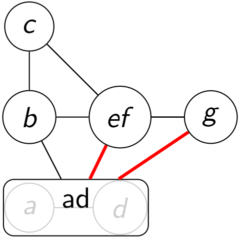
Contraction sequence and twin-width



Maximum red degree = 2

overall maximum red degree = 2

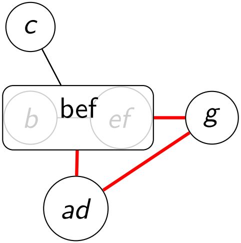
Contraction sequence and twin-width



Maximum red degree = 2

overall maximum red degree = 2

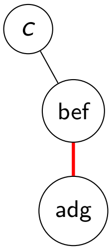
Contraction sequence and twin-width



Maximum red degree = 2

overall maximum red degree = 2

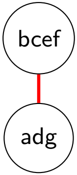
Contraction sequence and twin-width



Maximum red degree = 1

overall maximum red degree = 2

Contraction sequence and twin-width



Maximum red degree $= 1$

overall maximum red degree = 2

Contraction sequence and twin-width

Maximum red degree = 0

overall maximum red degree = 2

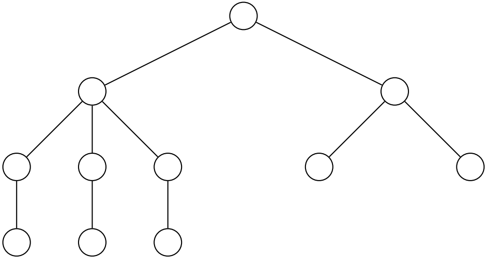
Contraction sequence and twin-width

Sequence of 2-contractions or 2-sequence, twin-width at most 2

Maximum red degree = 0

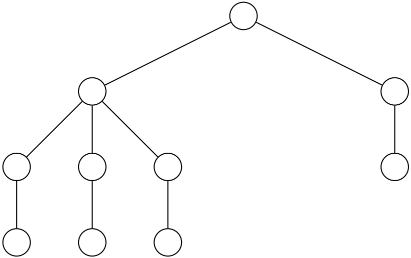
overall maximum red degree = 2

Graphs with bounded twin-width – trees



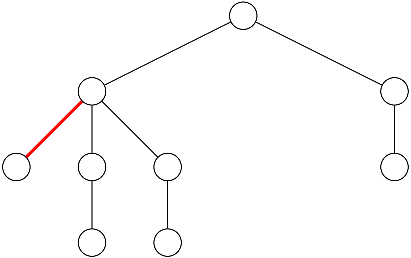
If possible, contract two twin leaves

Graphs with bounded twin-width – trees



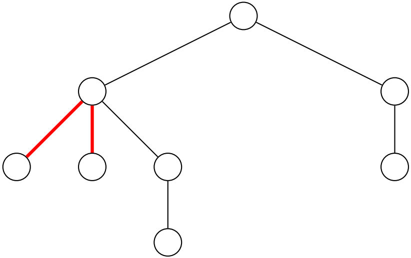
If not, contract a deepest leaf with its parent

Graphs with bounded twin-width – trees



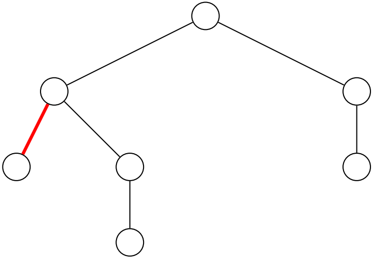
If not, contract a deepest leaf with its parent

Graphs with bounded twin-width – trees



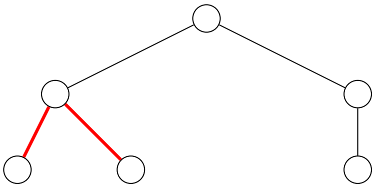
If possible, contract two twin leaves

Graphs with bounded twin-width – trees



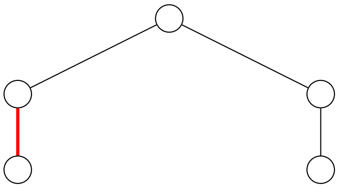
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



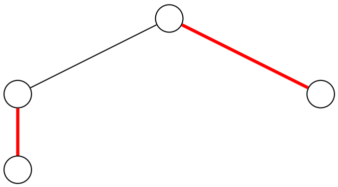
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



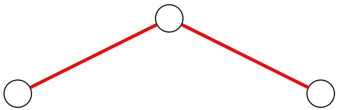
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



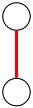
Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees



Cannot create a red degree-3 vertex

Graphs with bounded twin-width – trees

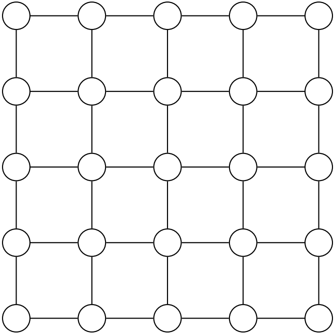


Cannot create a red degree-3 vertex

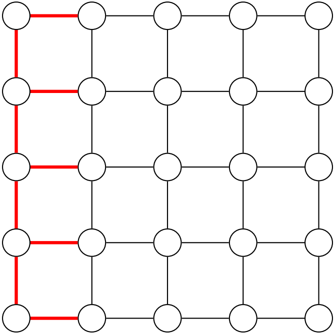
Graphs with bounded twin-width – trees

Generalization to bounded treewidth and even bounded rank-width

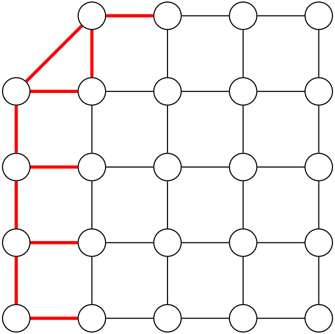
Graphs with bounded twin-width – grids



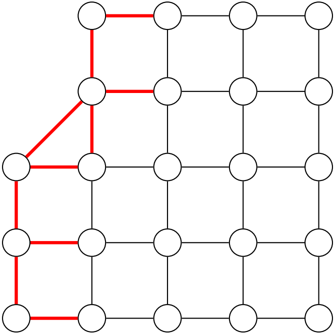
Graphs with bounded twin-width – grids



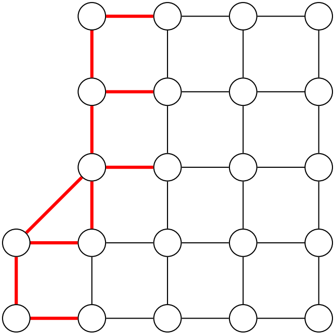
Graphs with bounded twin-width – grids



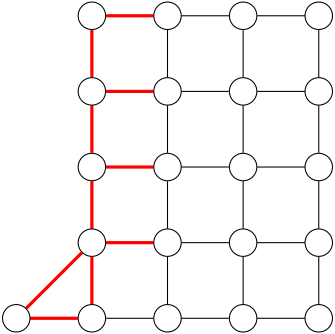
Graphs with bounded twin-width – grids



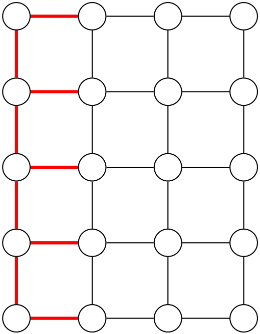
Graphs with bounded twin-width – grids



Graphs with bounded twin-width – grids



Graphs with bounded twin-width – grids



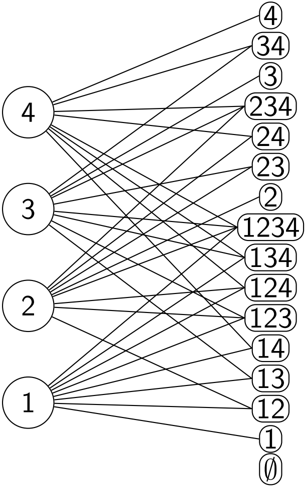
4-sequence for planar grids, 3d-sequence for d-dimensional grids

Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.

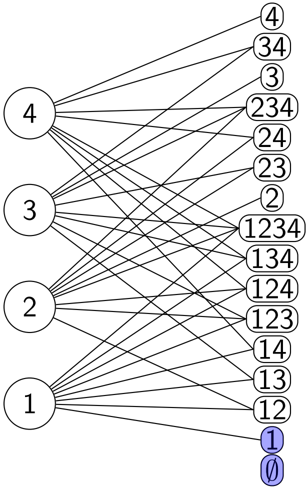
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



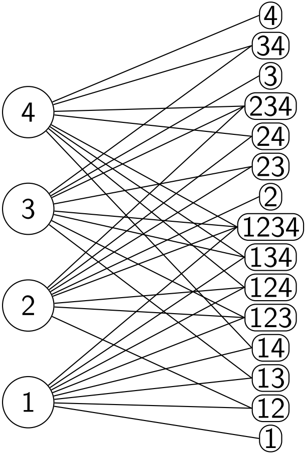
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



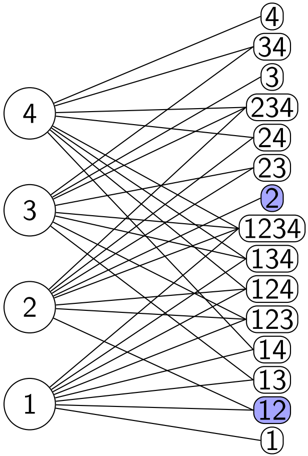
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



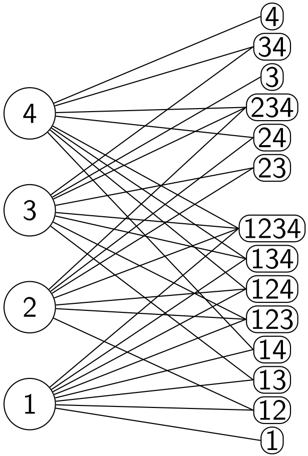
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



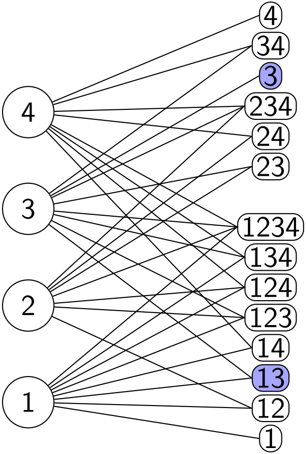
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



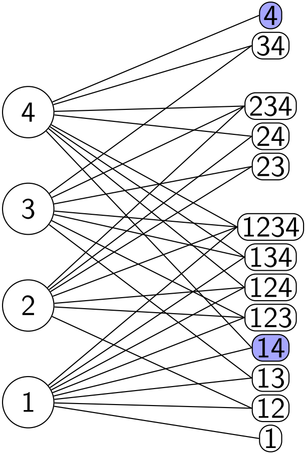
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



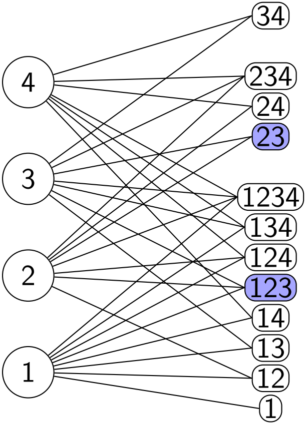
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



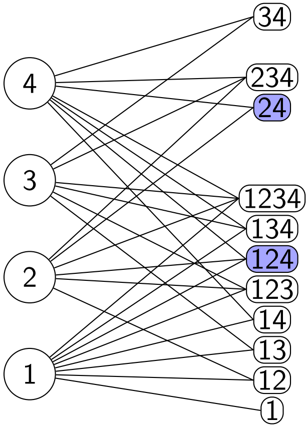
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



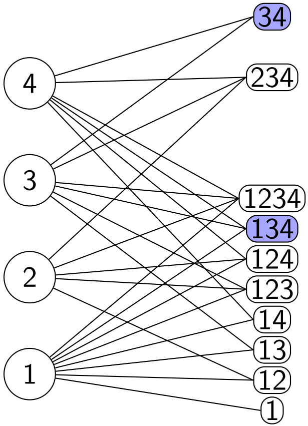
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



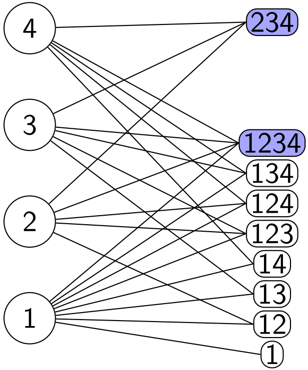
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



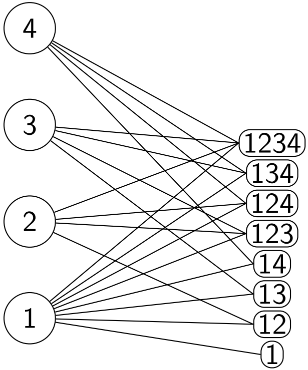
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



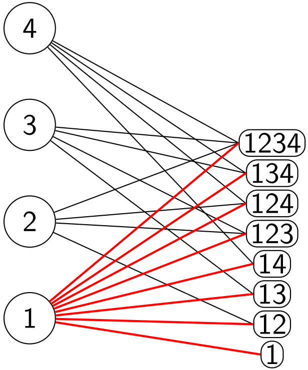
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



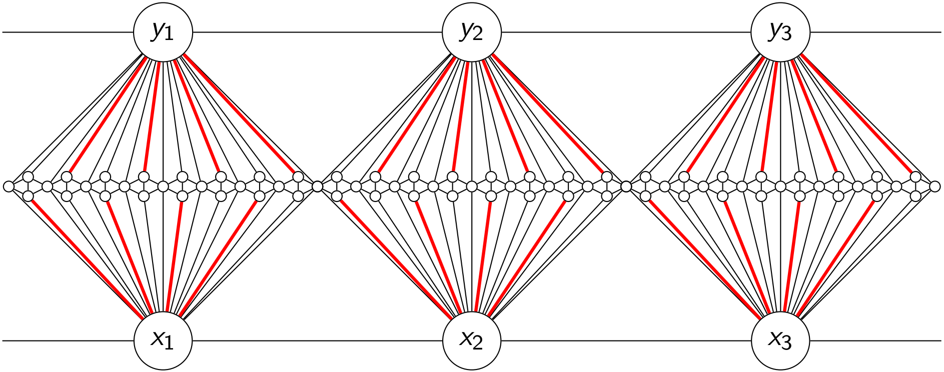
Universal bipartite graph

No $O(1)$-contraction sequence:
twin-width is *not* an iterated identification of near twins.



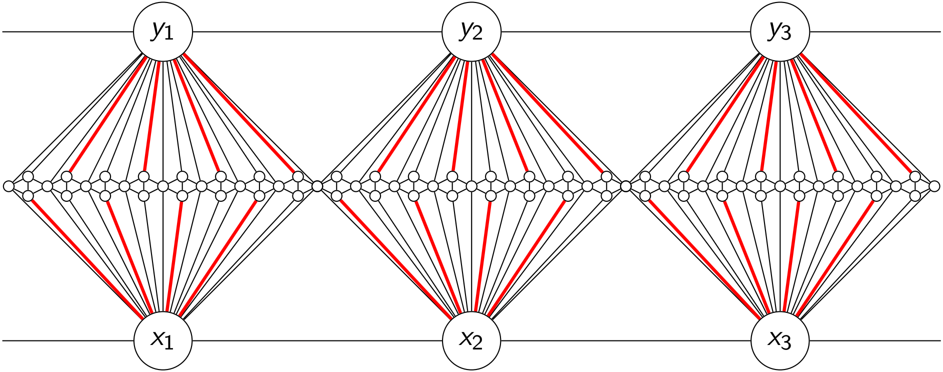
Graphs with bounded twin-width – planar graphs?

Graphs with bounded twin-width – planar graphs?



For every d, a planar trigraph without planar d-contraction

Graphs with bounded twin-width – planar graphs?



For every d, a planar trigraph without planar d-contraction

More powerfool tool needed

Mixed minor

Mixed zone: not horizontal nor vertical

$$\left[\begin{array}{cc|ccc|ccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]$$

3-mixed minor

Mixed minor

Mixed zone: not horizontal nor vertical

$$\left[\begin{array}{cc|ccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

3-mixed minor

A matrix is said t-**mixed free** if it does not have a t-mixed minor

Grid minor theorem for twin-width

Theorem (B, Kim, Thomassé, Watrigant 20)

If $\exists\sigma$ *s.t.* $Adj_\sigma(G)$ *is* t*-mixed free, then* $tww(G) = 2^{2^{O(t)}}$.

Grid minor theorem for twin-width

Theorem (B, Kim, Thomassé, Watrigant 20)

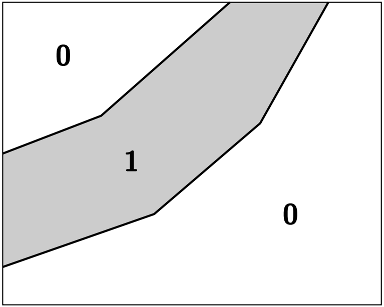
If $\exists\sigma$ s.t. $Adj_\sigma(G)$ is t-mixed free, then $tww(G) = 2^{2^{O(t)}}$.

Now to bound the twin-width of a class $\mathcal{C}$:

1) Find a *good* vertex-ordering procedure

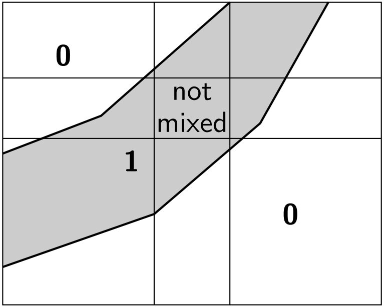
2) Argue that, in this order, a t-mixed minor would conflict with $\mathcal{C}$

Bounded twin-width – unit interval graphs



order by left endpoints

Bounded twin-width – unit interval graphs

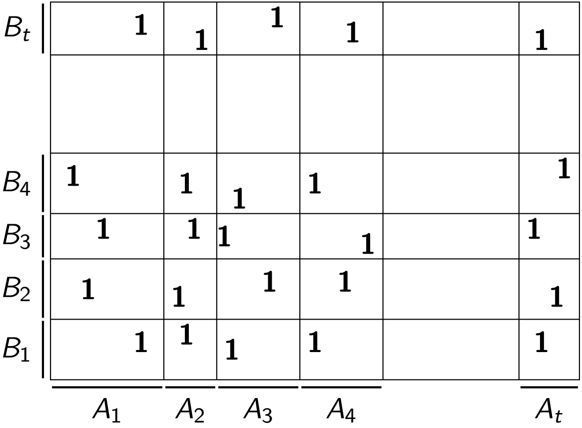


No 3-by-3 grid has all 9 cells crossed by two non-decreasing curves

Bounded twin-width – K_t-minor free graphs

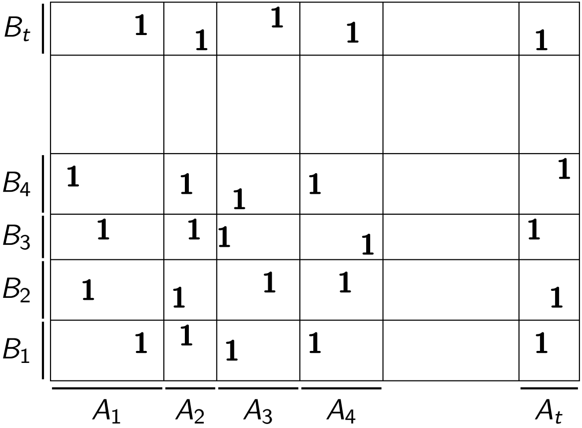
Given a hamiltonian path, we would just use this order

Bounded twin-width – K_t-minor free graphs



Contracting the $2t$ subpaths yields a $K_{t,t}$-minor, hence a K_t-minor

Bounded twin-width – K_t-minor free graphs



Instead we use a specially crafted lex-DFS discovery order

Theorem

The following classes have bounded twin-width, and $O(1)$-sequences can be computed in polynomial time.

- *Bounded rank-width, and even, boolean-width graphs,*
- *every hereditary proper subclass of permutation graphs,*
- *posets of bounded antichain size (seen as digraphs),*
- *unit interval graphs,*
- *K_t-minor free graphs,*
- *map graphs,*
- *subgraphs of d-dimensional grids,*
- *K_t-free unit d-dimensional ball graphs,*
- *$\Omega(\log n)$-subdivisions of all the n-vertex graphs,*
- *cubic expanders defined by iterative random 2-lifts from K_4,*
- *strong products of two bounded twin-width classes, one with bounded degree, etc.*

Theorem

The following classes have bounded twin-width, and $O(1)$-sequences can be computed in polynomial time.

- *Bounded rank-width, and even, boolean-width graphs,*
- *every hereditary proper subclass of permutation graphs,*
- *posets of bounded antichain size (seen as digraphs),*
- *unit interval graphs,*
- K_t*-minor free graphs,*
- *map graphs,*
- *subgraphs of d-dimensional grids,*
- K_t*-free unit d-dimensional ball graphs,*
- $\Omega(\log n)$*-subdivisions of all the n-vertex graphs,*
- *cubic expanders defined by iterative random 2-lifts from K_4,*
- *strong products of two bounded twin-width classes, one with bounded degree, etc.*

Can we solve problems faster, given an $O(1)$-sequence?

First-order model checking on graphs

GRAPH FO MODEL CHECKING	**Parameter:** $|\varphi|$

Input: A graph G and a first-order sentence $\varphi \in FO(\{E_2, =_2\})$
Question: $G \models \varphi$?

First-order model checking on graphs

GRAPH FO MODEL CHECKING	**Parameter:** $\lvert\varphi\rvert$
Input: A graph G and a first-order sentence $\varphi \in FO(\{E_2, =_2\})$	
Question: $G \models \varphi$?	

Example:

$$\varphi = \exists x_1 \exists x_2 \cdots \exists x_k \forall x \bigvee_{1 \leqslant i \leqslant k} x = x_i \vee \bigvee_{1 \leqslant i \leqslant k} E(x, x_i) \vee E(x_i, x)$$

$G \models \varphi$? $\Leftrightarrow$

First-order model checking on graphs

GRAPH FO MODEL CHECKING	**Parameter:** $\lvert\varphi\rvert$
Input: A graph G and a first-order sentence $\varphi \in FO(\{E_2, =_2\})$	
Question: $G \models \varphi$?	

Example:

$$\varphi = \exists x_1 \exists x_2 \cdots \exists x_k \forall x \bigvee_{1 \leqslant i \leqslant k} x = x_i \vee \bigvee_{1 \leqslant i \leqslant k} E(x, x_i) \vee E(x_i, x)$$

$G \models \varphi$? $\Leftrightarrow$ k-DOMINATING SET

First-order model checking on graphs

GRAPH FO MODEL CHECKING	**Parameter:** $\|\varphi\|$
Input: A graph G and a first-order sentence $\varphi \in FO(\{E_2, =_2\})$	
Question: $G \models \varphi$?	

Example:

$$\varphi = \exists x_1 \exists x_2 \cdots \exists x_k \bigwedge_{1 \leqslant i < j \leqslant k} \neg(x_i = x_j) \wedge \neg E(x_i, x_j) \wedge \neg E(x_j, x_i)$$

$G \models \varphi$? $\Leftrightarrow$

First-order model checking on graphs

Graph FO Model Checking **Parameter:** $|\varphi|$
Input: A graph G and a first-order sentence $\varphi \in FO(\{E_2, =_2\})$
Question: $G \models \varphi$?

Example:

$$\varphi = \exists x_1 \exists x_2 \cdots \exists x_k \bigwedge_{1 \leqslant i < j \leqslant k} \neg(x_i = x_j) \wedge \neg E(x_i, x_j) \wedge \neg E(x_j, x_i)$$

$G \models \varphi$? $\Leftrightarrow$ k-Independent Set

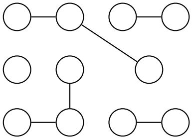
FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

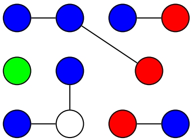
FO transduction: color by $O(1)$ unary relations, interpret, delete



FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

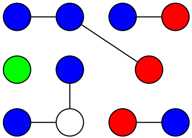
FO transduction: color by $O(1)$ unary relations, interpret, delete



FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x,y) = \neg E(x,y)$ (complement)
$\varphi(x,y) = E(x,y) \vee \exists z E(x,z) \wedge E(z,y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete



$$\varphi(x,y) = E(x,y) \vee (G(x) \wedge B(y) \wedge \neg\exists z R(z) \wedge E(y,z))$$
$$\vee (R(x) \wedge B(y) \wedge \exists z R(z) \wedge E(y,z) \wedge \neg\exists z B(z) \wedge E(y,z))$$

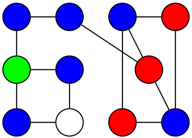
FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula

$\varphi(x, y) = \neg E(x, y)$ (complement)

$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete

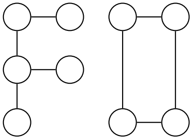


$$\varphi(x, y) = E(x, y) \vee (G(x) \wedge B(y) \wedge \neg\exists z R(z) \wedge E(y, z))$$
$$\vee(R(x) \wedge B(y) \wedge \exists z R(z) \wedge E(y, z) \wedge \neg\exists z B(z) \wedge E(y, z))$$

FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

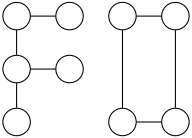
FO transduction: color by $O(1)$ unary relations, interpret, delete



FO interpretations and transductions

FO interpretation: redefine the edges by a first-order formula
$\varphi(x, y) = \neg E(x, y)$ (complement)
$\varphi(x, y) = E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$ (square)

FO transduction: color by $O(1)$ unary relations, interpret, delete



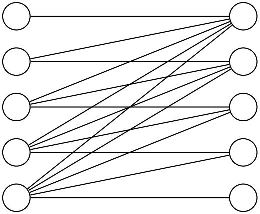
Theorem (B, Kim, Thomassé, Watrigant '20)

Bounded twin-width is preserved by transduction.

Monadically Stable and NIP

Stable class: no transduction of the class contains all ladders
NIP class: no transduction of the class contains all graphs

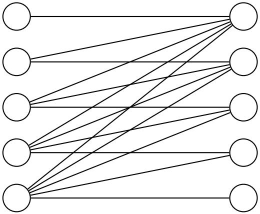
ladder



Monadically Stable and NIP

Stable class: no transduction of the class contains all ladders
NIP class: no transduction of the class contains all graphs

ladder

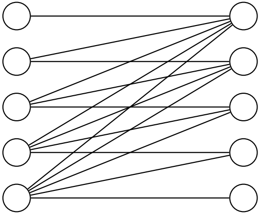


Bounded-degree graphs $\rightarrow$ stable
Unit interval graphs $\rightarrow$ NIP but not stable
Interval graphs $\rightarrow$ not NIP

Monadically Stable and NIP

Stable class: no transduction of the class contains all ladders
NIP class: no transduction of the class contains all graphs

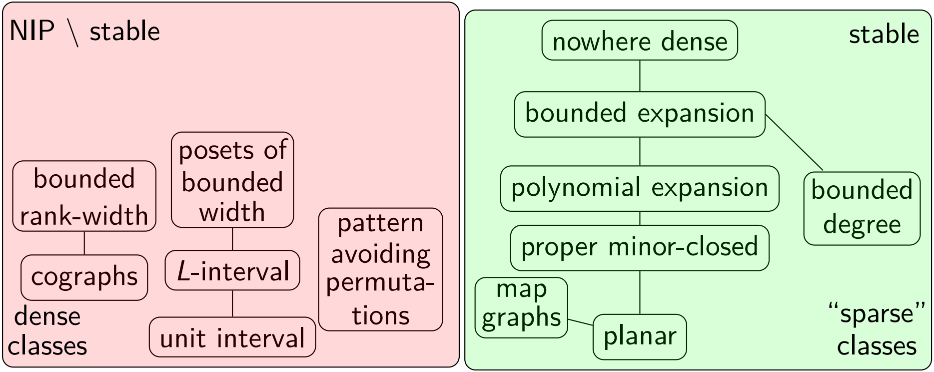
ladder



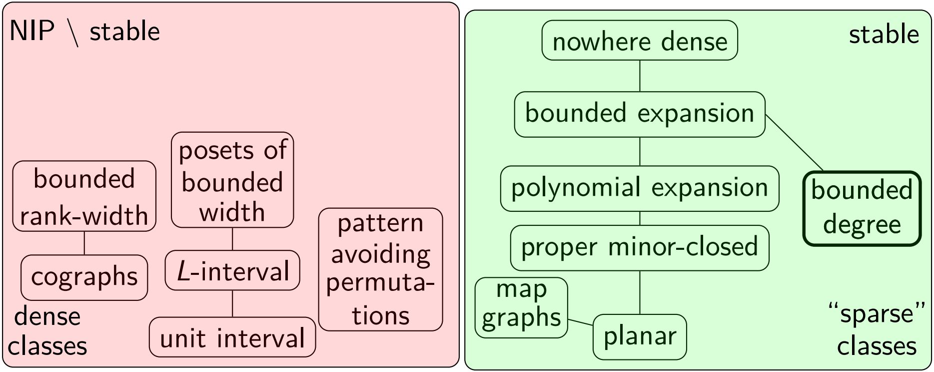
Bounded-degree graphs $\rightarrow$ stable
Unit interval graphs $\rightarrow$ NIP but not stable
Interval graphs $\rightarrow$ not NIP

Bounded twin-width classes $\rightarrow$ NIP but not stable in general

Classes with known tractable FO model checking

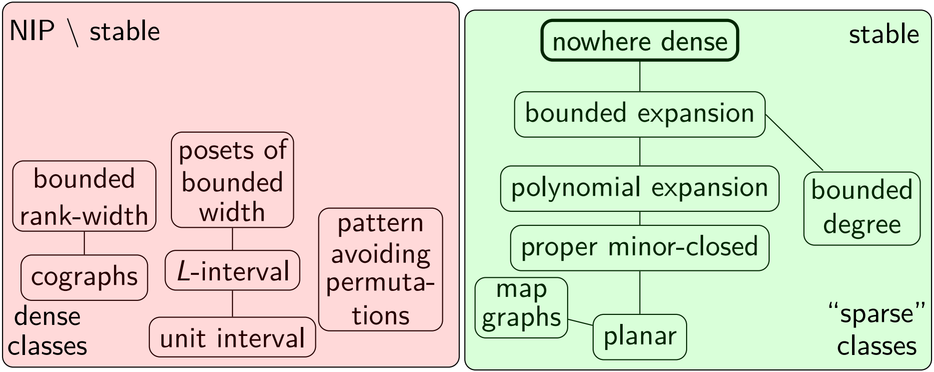


Classes with known tractable FO model checking



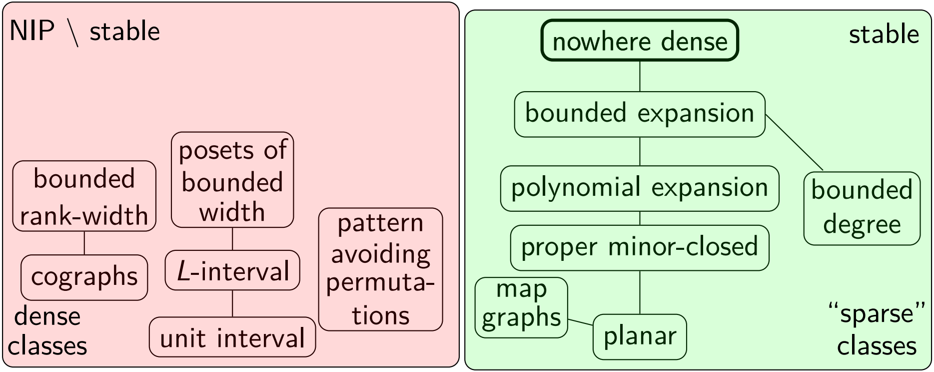
FO MODEL CHECKING solvable in $f(|\varphi|)n$ on bounded-degree graphs [Seese '96]

Classes with known tractable FO model checking



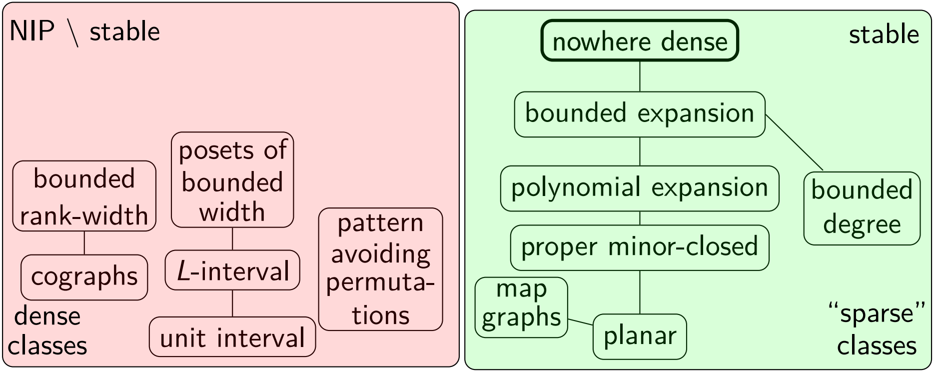
FO MODEL CHECKING solvable in $f(|\varphi|)n^{1+\varepsilon}$ on any nowhere dense class [Grohe, Kreutzer, Siebertz '14]

Classes with known tractable FO model checking



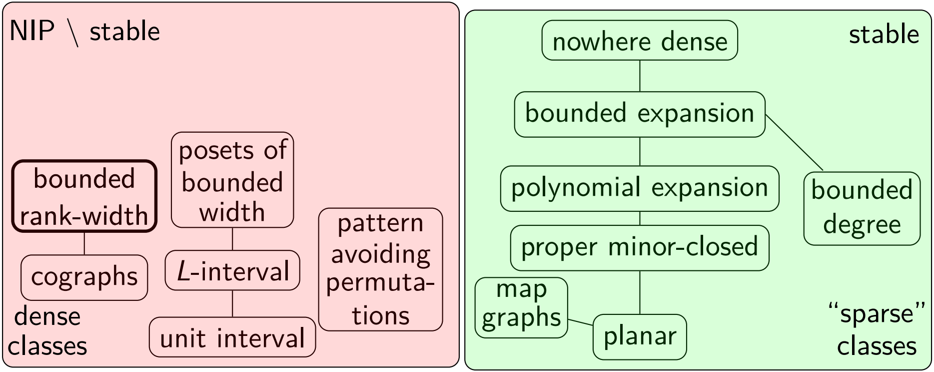
End of the story for the subgraph-closed classes
tractable FO MODEL CHECKING ⇔ nowhere dense ⇔ stable

Classes with known tractable FO model checking



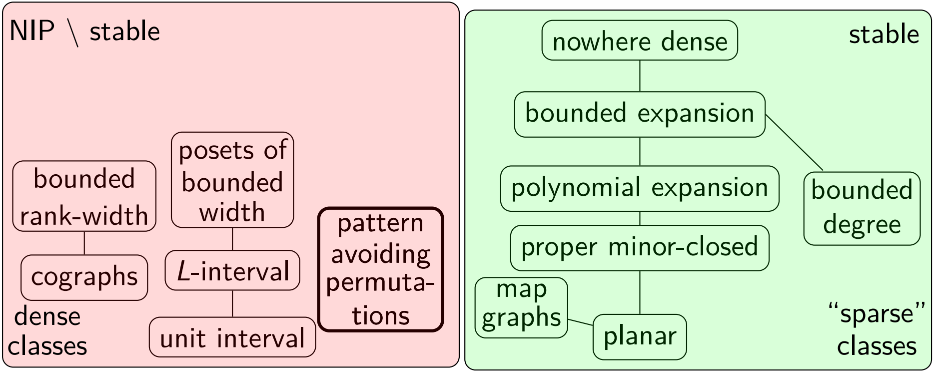
New program: transductions of nowhere dense classes
Not sparse anymore but still stable

Classes with known tractable FO model checking



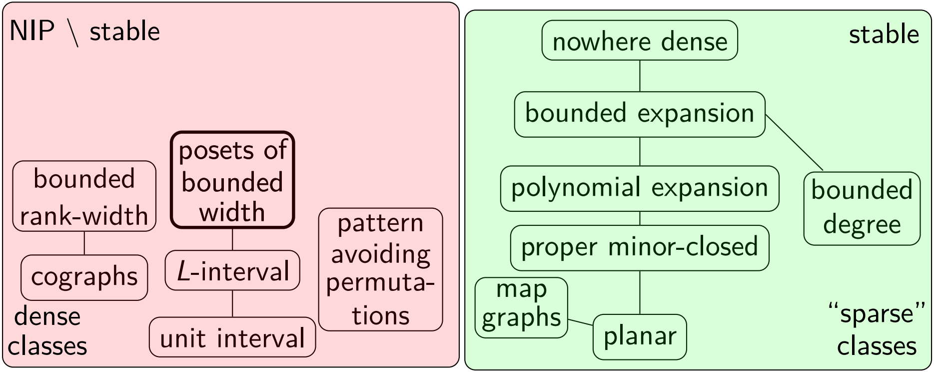
MSO_1 Model Checking solvable in $f(|\varphi|, w)n$ on graphs of rank-width w
[Courcelle, Makowsky, Rotics '00]

Classes with known tractable FO model checking



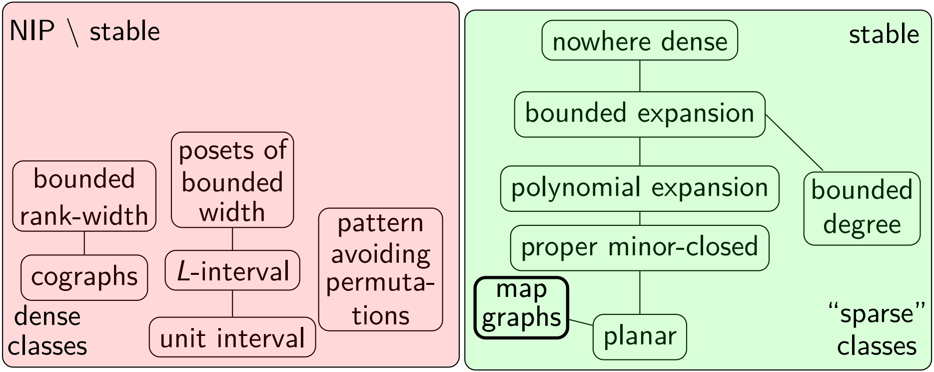
Is σ a subpermutation of τ? solvable in $f(|\sigma|)|\tau|$
[Guillemot, Marx '14]

Classes with known tractable FO model checking



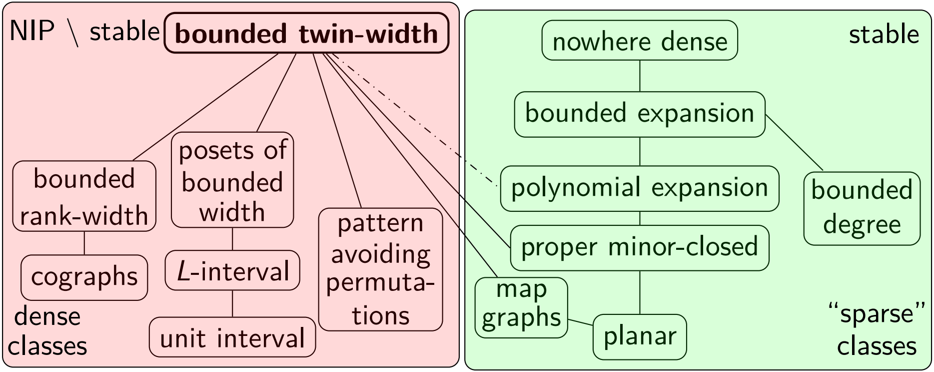
FO MODEL CHECKING solvable in $f(|\varphi|, w)n^2$ on posets of width w
[GHLOORS '15]

Classes with known tractable FO model checking



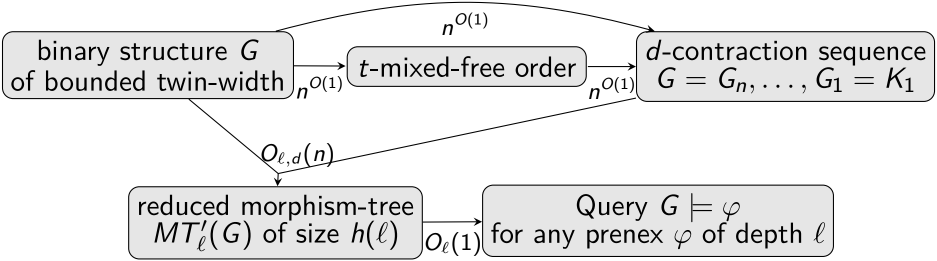
FO Model Checking solvable in $f(|\varphi|)n^{O(1)}$ on map graphs
[Eickmeyer, Kawarabayashi '17]

Classes with known tractable FO model checking

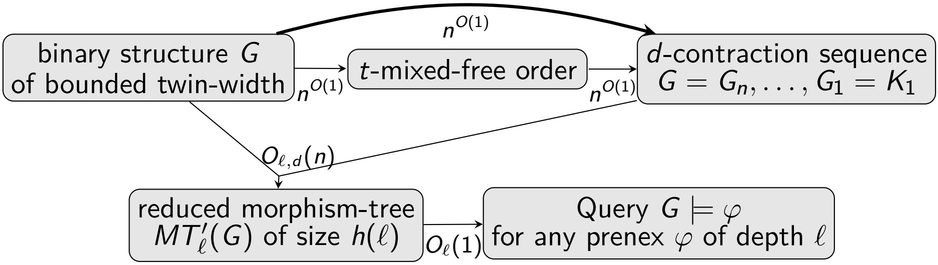


FO Model Checking solvable in $f(|\varphi|, d)n$ on graphs with a d-sequence [B, Kim, Thomassé, Watrigant '20]

Workflow of the FO model checking algorithm

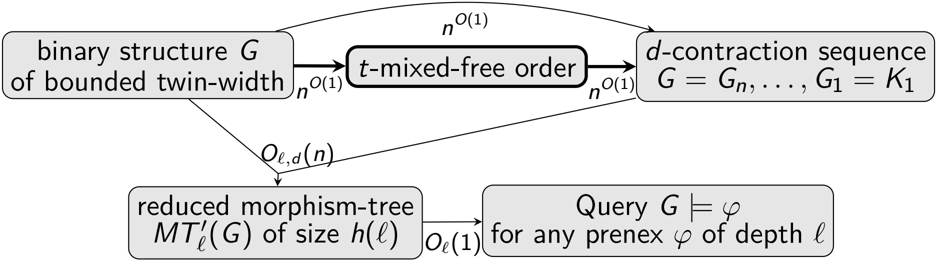


Workflow of the FO model checking algorithm



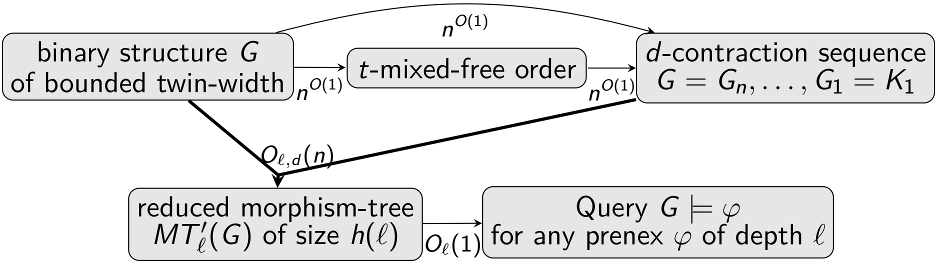
Direct examples: **trees**, bounded rank-width, **grids**, d-dimensional grids, K_t-free unit ball graphs

Workflow of the FO model checking algorithm



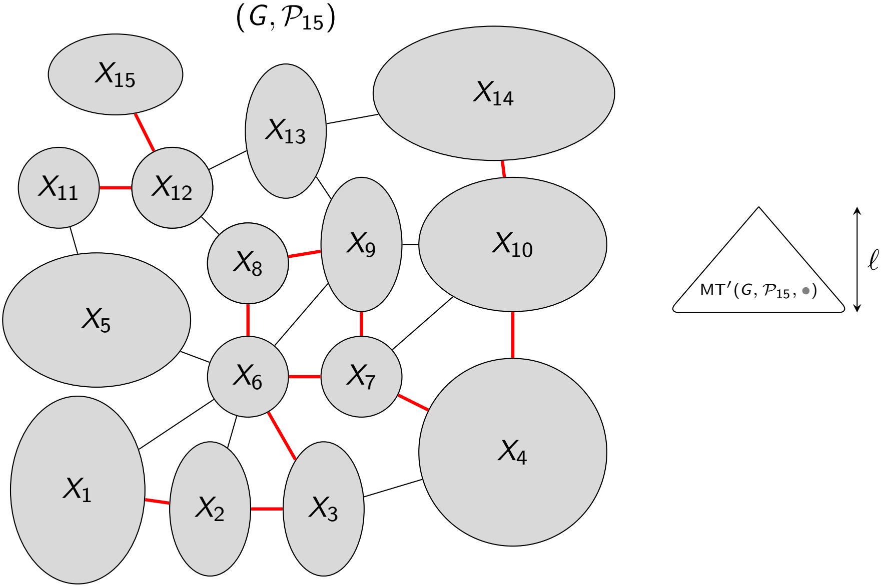
Detour via mixed minor for: pattern-avoiding permutations, bounded width posets, K_t-**minor free graphs**

Workflow of the FO model checking algorithm

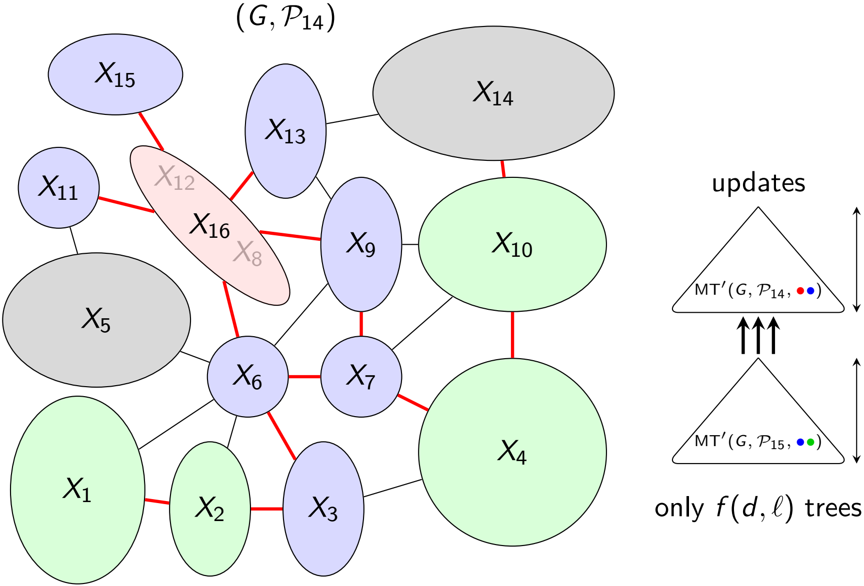


Let us see a snapshot of the FO model checking

DP for FO model checking with d-sequence



DP for FO model checking with d-sequence



Open questions

Algorithm to compute/approximate twin-width in general
Fully classify classes with tractable FO model checking

On arxiv
Twin-width I: tractable FO model checking [BKTW '20]
Twin-width II: small classes [BGKTW '20]
Twin-width III: Max Independent Set and Coloring [BGKTW '20]