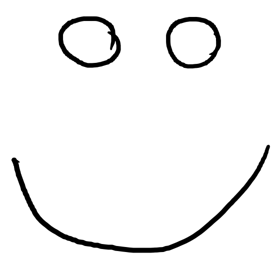
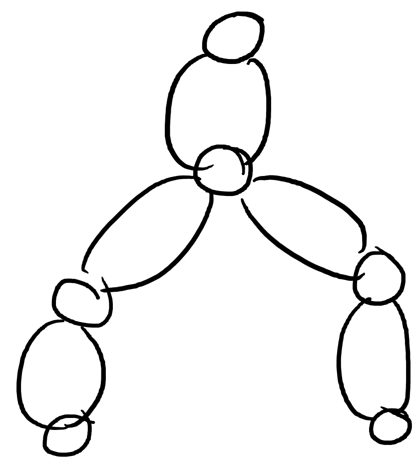
Vérification de formule FO sur graphes via la paramétrisation twin-width

Bonnet, Édouard, et al. "Twin-width I: tractable FO model checking." *FOCS 2020*.

Theorem:

One can decide whether $G \vDash \phi$ in time $f(d, |\phi|) \cdot n$, where $\boldsymbol{d = twinwidth(G)}$ and f is some computable function.





CONF	Year	Paper Title	Key Authors	Core Contribution
FOCS	2020	Twin-width I: tractable FO model checking	É. Bonnet, E. J. Kim, S. Thomassé, R. Watrigan	Introduces the twin-width parameter; establishes FPT FO model checking given a witness; proves preservation under FO transductions.
SODA	2021	Twin-width II: small classes	É. Bonnet, C. Geniet, E. J. Kim, S. Thomassé, R. Watrigant	Proves bounded twin-width classes are combinatorially small (growth); establishes -adjacency labeling scheme.
SODA	2022	Twin-width VI: the lens of contraction sequences	É. Bonnet, E. J. Kim, A. Reinald, S. Thomassé	Unifies classical width measures via restricted contraction sequences; introduces oriented twin-width.
STOC	2022	Twin-width IV: ordered graphs and matrices	É. Bonnet, U. Giocanti, P. Ossona de Mendez, P. Simon, S. Thomassé, S. Toruńczyk	Settles the Balogh-Bollobás-Morris conjecture; proves the equivalence of monadic NIP, FPT FO model checking, and small growth in ordered graphs.
SODA	2023	Fixed-parameter tractability of DIRECTED MULTICUT with three terminal pairs parameterized by the size of the cutset: twin-width meets flow-augmentation	M. Hatzel, L. Jaffke, P. T. Lima, T. Masařík, M. Pilipczuk, R. Sharma, M. Sorge	Solves Directed Multicut for by translating directed flow-augmentation CSPs into matrix-based FO model checking.
SODA	2023	Twin-width of random graphs	M. Bonamy, É. Bonnet, H. Déprés, L. Esperet, C. Geniet, M. Hatzel, S. Thomassé, R. Watrigant	Analyzes the asymptotic behavior of twin-width in Erdős-Rényi random graphs , establishing sharp threshold bounds.
FOCS	2023	Graphs of bounded twin-width are quasi-polynomially -bounded	T. Korhonen, M. Pilipczuk, et al.	Establishes the first quasi-polynomial chromatic bounding function for bounded twin-width graph families.
FOCS	2023	Flip-width: Cops and Robber on dense graphs	S. Toruńczyk, et al.	Introduces flip-width via flipper games to successfully unify twin-width theory and sparsity theory.
FOCS	2024	An Optimal Algorithm for Sorting Pattern-Avoiding Sequences	Michal Opler	Presents a linear-time sorting algorithm for pattern-avoiding sequences, yielding linear-time sorting of bounded twin-width permutations.
SODA	2025	Bounding -scatter dimension via metric sparsity	R. Bourneuf, M. Pilipczuk	Investigates metric properties and VC-density in bounded twin-width graph families via scatter dimension bounds.
STOC	2025	Reduced bandwidth: A qualitative strengthening of twin-width in minor-closed classes (and beyond)	É. Bonnet, O. Kwon, D. R. Wood	Formulates reduced bandwidth to enforce strict topological layouts on contraction sequences in minor-closed classes.
STOC	2025	Merge-width and first-order model checking	J Dreier, S Toruńczyk	Proposes a parameter that generalizes twin-width

First order (FO) formulas on graphs

- $\neg \ \wedge \ \vee \ \Rightarrow$
- $\forall v \cdot (\dots) \qquad \exists v \cdot (\dots)$
- Universe = set of vertices of a given graph G
  - Variables = individual vertices
  - Only quantification on vertices (no subsets, relations, etc)

Adjacency predicate:

- **adj(u, v)** is true is and only if $uv \in E(G)$, i.e., uv is an edge
  - $adj(u, u)$ always false, $adj(u, v) \Leftrightarrow adj(v, u)$

Also, equality predicates $u = v$ and $u \neq v$

$G \vDash \phi$ if FO formula ϕ is true on G (G is a model for ϕ)

Deciding if $G \vDash \phi$ is PSPACE-complete

Time complexity of $n^{O(|\phi|)}$ possible. $n = |V(G)|$

$|\phi|$ = size of the formula = number of variables

Deciding if $G \vDash \phi$ is PSPACE-complete

Time complexity of $n^{O(|\phi|)}$ possible.

$|\phi|$ = size of the formula = number of variables

Too slow! Parameterized complexity would be better.

e.g., $2^{|\phi|}n$, or $|\phi|^{|\phi|}n$, or anything $f(|\phi|)n$

Deciding if $G \vDash \phi$ is PSPACE-complete

Time complexity of $n^{O(|\phi|)}$ possible.

$|\phi|$ = size of the formula = number of variables

Too slow! Parameterized complexity would be better.

e.g., $2^{|\phi|}n$, or $|\phi|^{|\phi|}n$, or anything $f(|\phi|)n$

Impossible under the Exponential Time Hypothesis.

Deciding if $G \vDash \phi$ is PSPACE-complete

Time complexity of $n^{O(|\phi|)}$ possible.

$|\phi|$ = size of the formula = number of variables

Too slow! Parameterized complexity would be better.

e.g., $2^{|\phi|}n$, or $|\phi|^{|\phi|}n$, or anything $f(|\phi|)n$

Impossible under the Exponential Time Hypothesis.

Idea: take a structural parameter d of the graph.

Aim for $f(d, |\phi|) \cdot n$.

Example: $d =$ max degree of G.

$N(v) =$ neighbors of v

$maxdeg(G) = max_{v \in V(G)} |N(v)|$

k-Clique(G) #returns true if G has a clique of size k

Example: d = max degree of G.

$N(v)$ = neighbors of v

$maxdeg(G) = max_{v \in V(G)} |N(v)|$

```
k-Clique(G)          #returns true if G has a clique of size k
  for v ∈ V(G)
    for X ⊆ N(v) ∪ {v}
      if X is a clique of size k
        return true
  return false
```

Theorem 1:

One can decide whether $G \vDash \phi$ in time $f(d, |\phi|) \cdot n$, where $d = maxdeg(G)$ and f is some computable function.

Uses *Gaifman normal form*.

Theorem 2:

Any FO formula ϕ on graphs can be expressed as an equivalent formula in Gaifman normal form:

$$\phi \equiv \exists x_1, x_2, \ldots, x_k \cdot \left(\bigwedge_{i<j} \left(dist_G(x_i, x_j) > 2r \right) \wedge \bigwedge_i \psi(x_i) \right)$$

where

- $dist_G(x_i, x_j) = \text{distance in } G$
- k, r depend only on $|\phi|$
- $\psi(x_i)$ only depends on vertices at distance at most r from x_i
- On graphs of max degree d, only $O(rd^r)$ such vertices.
- => $O(rd^r)$ TYPES of vertices, makes it easy.

Theorem 1:

One can decide whether $G \vDash \phi$ in time $f(d, |\phi|) \cdot n$, where $d = maxdeg(G)$ and f is some computable function.

Uses *Gaifman normal form*.

Theorems:

FO model checking can be done in time $f(d, |\phi|)poly(n)$ if :

- d = maximum degree
- d = tree-width
- d = clique-width
- d = poset-width

Theorems:

FO model checking can be done in time $f(d, |\phi|)poly(n)$ if :

- d = maximum degree
- d = tree-width
- d = clique-width
- d = poset-width

FO model checking can be done in time $f(|\phi|)poly(n)$ on:

- planar graphs
- interval graphs
- permutation graphs
- other meta-classes (bounded expansion graphs, proper minor-closed graphs, H minor-free graphs, ...)

Theorems:

FO model checking can be done in time $f(d, |\phi|)poly(n)$ if :

- d = maximum degree
- d = tree-width
- d = clique-width
- d = poset-width

FO model checking can be done in time $f(|\phi|)poly(n)$ on:

- planar graphs
- interval graphs
- permutation graphs
- other meta-classes (bounded expansion graphs, proper minor-closed graphs, H minor-free graphs, ...)

Graphs of bounded **twin-width**

Theorem 3:

One can decide whether $G \vDash \phi$ in time $f(d, |\phi|) \cdot n$, where $\boldsymbol{d = twinwidth(G)}$ and f is some computable function.

Let $X, Y \subseteq V(G)$ be disjoint.

A link between X and Y is:

- **black** if every possible edge between X and Y is present in G
- **white** if there is no edge between X and Y in G.
- **red** otherwise (disagreement between X and Y).

Vertex contractions

- Each vertex u is labeled by a subset $l(u) \subseteq V(G)$.
- Initially, each $u \in V(G)$ is labeled by $l(u) = \{u\}$.
- Contraction of two labeled vertices u and v:
  1. Remove u and v from G.
  2. Add a new vertex w with label $l(w) = l(u) \cup l(v)$.
  3. Add a link of the appropriate color between w and every other vertex, using labels as X and Y.
- **Red degree** of a vertex = number of red links incident to the vertex.

Twin-width

- Contraction sequence of G = ordered list of contractions that results in a single vertex.
- **Red degree** of a contraction sequence = maximum red degree of a vertex encountered during the whole sequence.
- $tww(G)$ = minimum **red degree** among all contraction sequences of G.

Has triangle:

$$\exists x_1, x_2, x_3 \cdot (adj(x_1, x_2) \wedge adj(x_1, x_3) \wedge adj(x_2, x_3))$$

Warm-up: $tww(G) = 0$

Idea:during the contraction sequence, detect if some $l(v)$ contains a triangle at any point.

For each vertex v during the whole contraction sequence

$$A[v] = \begin{cases} 1 \text{ if } \text{ vertices in } l(v) \text{contain an edge} \\ 0 \text{ otherwise} \end{cases}$$

$$B[v] = \begin{cases} 1 \text{ if } \text{ vertices in } l(v) \text{contain a triangle} \\ 0 \text{ otherwise} \end{cases}$$

Has triangle:
$\exists x_1, x_2, x_3 \cdot (adj(x_1, x_2) \wedge adj(x_1, x_3) \wedge adj(x_2, x_3))$

$A[v] = 0, B[v] = 0$ for all $v \in V(G)$

for each contraction of u and v in the sequence

Replace u, v with $u + v$

Update $l(u + v) = l(u) \cup l(v)$, update colors

if $link(u, v) = \boldsymbol{black}$

$A[u + v] = 1$

$B[u + v] = A[u] \vee A[v]$

else if $link(u, v) = white$

$A[u + v] = A[u] \vee A[v]$

$B[u + v] = B[u] \vee B[v]$

Has triangle:

$$\exists x_1, x_2, x_3 \cdot (adj(x_1, x_2) \wedge adj(x_1, x_3) \wedge adj(x_2, x_3))$$

Warm-up 2: $tww(G) = 1$

For each vertex v during the whole contraction sequence

$$A[v] = \begin{cases} 1 \text{ if } \text{ vertices in } l(v) \text{ contain an edge} \\ 0 \text{ otherwise} \end{cases}$$

$$B[v] = \begin{cases} 1 \text{ if } \text{ vertices in } l(v) \text{ contain a triangle} \\ 0 \text{ otherwise} \end{cases}$$

Has triangle:

$$\exists x_1, x_2, x_3 \cdot (adj(x_1, x_2) \wedge adj(x_1, x_3) \wedge adj(x_2, x_3))$$

Warm-up 2: $tww(G) = 1$

For each vertex v during the whole contraction sequence

$$A[v] = \begin{cases} 1 \text{ if } \text{ vertices in } l(v) \text{ contain an edge} \\ 0 \text{ otherwise} \end{cases}$$

$$B[v] = \begin{cases} 1 \text{ if } \text{ vertices in } l(v) \text{ contain a triangle} \\ 0 \text{ otherwise} \end{cases}$$

For each u, v that have a red link:

$C[u, v] = 1$ iff G has an edge ab with $a, b \in l(u)$, such that a, b have a common neighbor in $l(v)$

Updates not too complicated to do.

Has triangle:

$$\exists x_1, x_2, x_3 \cdot (adj(x_1, x_2) \wedge adj(x_1, x_3) \wedge adj(x_2, x_3))$$

Now for $\boldsymbol{tww(G) > 1}$.

Need info on larger set of nodes connected by red edges.

For each vertex v during the whole contraction sequence

$$A[v] = \begin{cases} 1 \text{ if vertices in } l(v) \text{ contain an edge} \\ 0 \text{ otherwise} \end{cases}$$

$$B[v] = \begin{cases} 1 \text{ if vertices in } l(v) \text{ contain a triangle} \\ 0 \text{ otherwise} \end{cases}$$

For each u, v that have a red link:

$C[u, v] = 1$ iff G has an edge ab with $a, b \in l(u)$, such that a, b have a common neighbor in $l(v)$

For each red triangle u, v, w

$D[u, v, w] = 1$ iff we can choose one vertex in each of $l(u), l(v), l(w)$ that all share an edge.

Has k-clique:

$$\exists x_1, \ldots, x_k \cdot \left(\bigwedge_{i<j} adj(x_i, x_j)\right)$$

For each subgraph H of G that is connected by **red edges** of diameter at most k

For each $f : V(H) \to \{0,1,\ldots,k\}$

$Z[H,f] = 1$ if G has a clique with $f(h)$vertices in each $h \in V(H)$

Z not insanely hard to update.

Want: $Z[H,f] = 1$ such that $f(h) = k$ for some $h \in V(H)$.

Generalization to $G \vDash \phi$

For two subgraphs vertices G_1, G_2

$TYPE(G_1) = TYPE(G_2)$ if $\forall\psi \cdot (G_1 \vDash \psi \Leftrightarrow G_2 \vDash \psi)$

where ψ ranges over formulas of size at most $|\phi|$

Generalization to $G \vDash \phi$

For two subgraphs vertices G_1, G_2

$TYPE(G_1) = TYPE(G_2)$ if $\forall \psi \cdot (G_1 \vDash \psi \Leftrightarrow G_2 \vDash \psi)$

where ψ ranges over formulas of size at most $|\phi|$

For each contraction of u, v into $u + v$

For each subgraph H of G that is connected by **red edges** of diameter at most k

Update $TYPE(H)$

Generalization to $G \vDash \phi$

For two subgraphs vertices G_1, G_2

$TYPE(G_1) = TYPE(G_2)$ if $\forall \psi \cdot (G_1 \vDash \psi \Leftrightarrow G_2 \vDash \psi)$

where ψ ranges over formulas of size at most $|\phi|$

For each contraction of u, v into $u + v$

For each subgraph H of G that is connected by **red edges** of diameter at most k

Update $TYPE(H)$

The number of encountered types is $f(tww, |\phi|)$

$TYPE(H)$ more insanely hard to update (but possible).

Want: A single contracted vertex whose type satisfies ϕ