Algorithmic Meta-Theorems for Distributed Computing

Pierre Fraigniaud

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE


L. Blin


F. Fomin


S. Gay


P. Golovach


P. Montealegre


I. Rapaport


I. Todinca

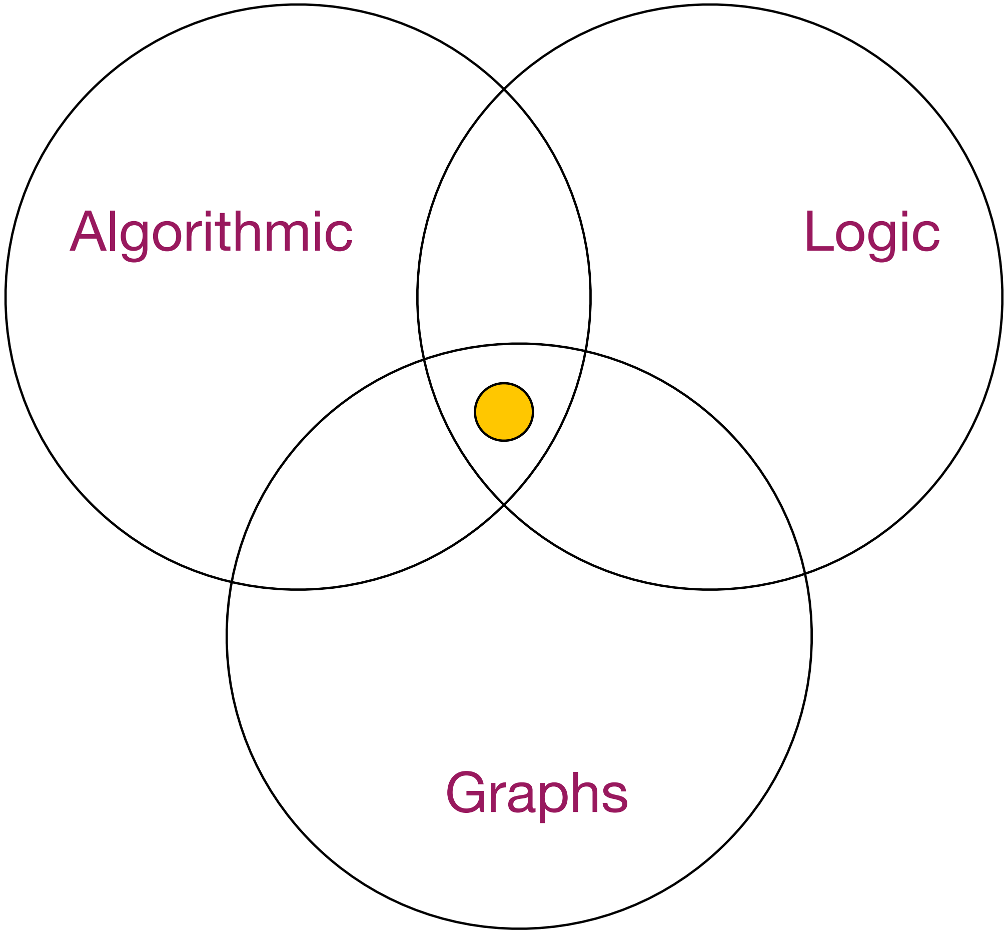
Workshop on Foundations of Distributed and Parallel Graph Algorithms, Venice, Italy, May 18-21, 2026

Algorithmic Meta-Theorems

Meta-Theorems

- **Theorems** are proved within a given formal system
- **Metatheorems** are statements about formal systems
- **Example:** Gödel's completeness theorem states that *first-order logic is complete*.

Algorithmic Meta-Theorems

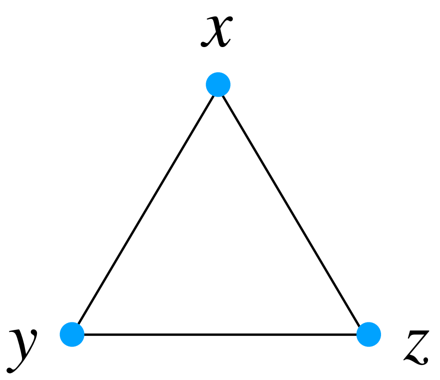


First-Order Logic on (labeled) Graphs

- Objects: Vertices
- Boolean predicates:
  - Equality: $x = y$
  - Adjacency: $\mathrm{adj}(x, y)$
  - Labels: $\mathrm{red}(x)$
- Quantifiers: $\forall \quad \exists$
- Connectors: $\neg \quad \wedge \quad \vee$

Example: triangle-freeness

$$\neg\Big(\exists x \exists y \exists z \; \big(\mathrm{adj}(x, y) \wedge \mathrm{adj}(y, z) \wedge \mathrm{adj}(z, x)\big)\Big)$$



Second-Order Logic on (labeled) Graphs

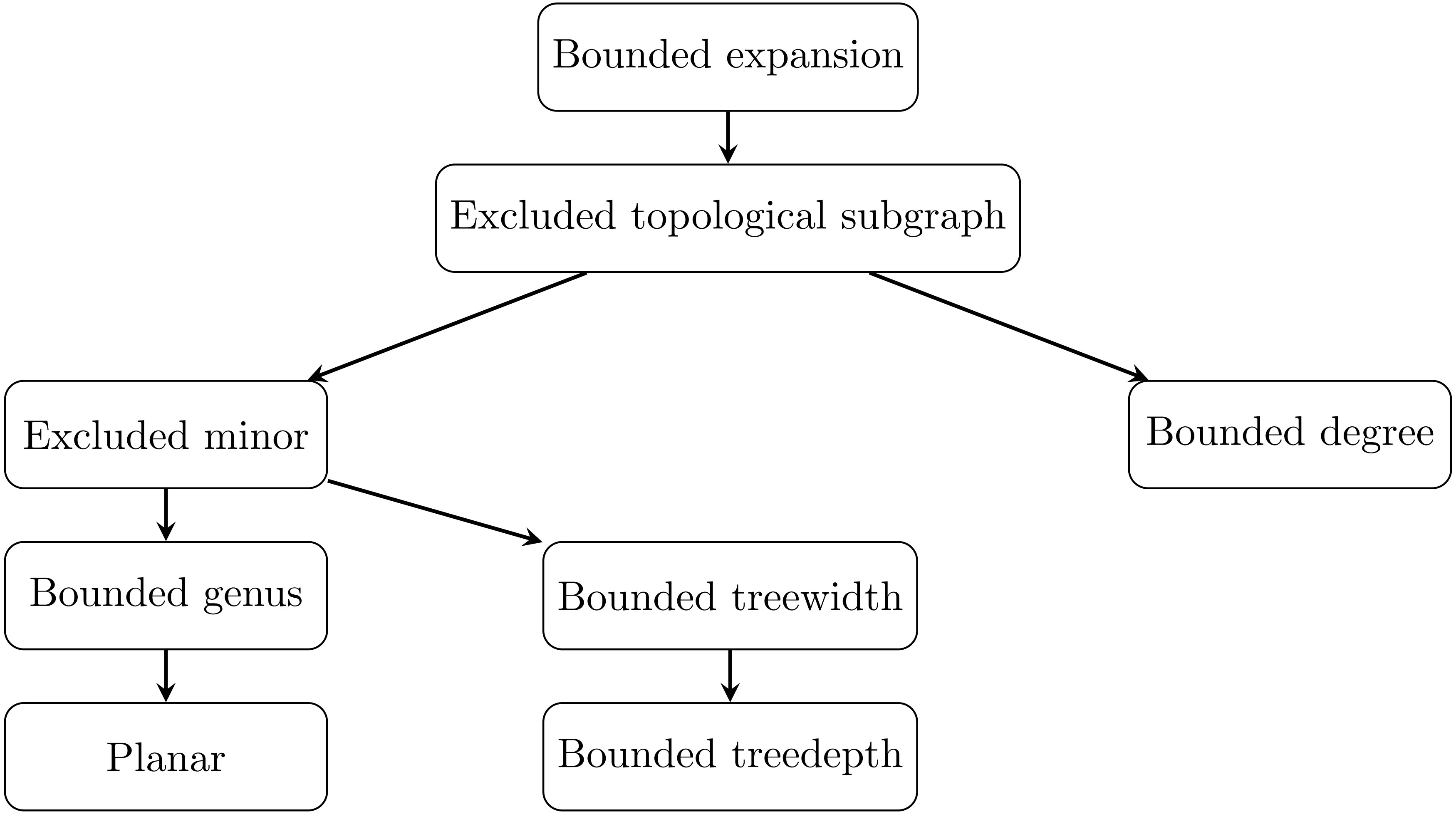
- Second-order logic quantifies over relations
- Monadic second-order logic (MSO): quantification is over sets.
- Example: Bipartiteness

$$\begin{aligned}\exists A\,\exists B\,\Big(&\big(\forall x\,(x \in A \land x \notin B) \lor (x \notin A \land x \in B)\big) \land \\ &\big(\forall x \forall y\ \neg\mathsf{adj}(x, y) \lor (x \in A \land y \in B) \lor (x \in B \land y \in A)\big)\Big)\end{aligned}$$

- Example: Path(x,y)

$$\begin{aligned}\exists S\Big(&\big(x \in S \land \exists z \in S\ \mathsf{adj}(x, z)\big) \land \big(y \in S \land \exists z \in S\ \mathsf{adj}(y, z)\big) \land \\ &\big(\forall z \in S \setminus \{x, y\}\ \exists w\,\exists w'\ \mathsf{adj}(z, w) \land \mathsf{adj}(z, w') \land w \neq w'\big)\Big)\end{aligned}$$

Graph Classes



Bounded expansion

Excluded topological subgraph

Excluded minor

Bounded degree

Bounded genus

Bounded treewidth

Planar

Bounded treedepth

Another Parameter: Clique-Width

- Grammar for constructing graphs
  - NewVertex_i : Creation of a new vertex with color i
  - $G \mid\mid H$: Disjoint union of two colored graphs G and H
  - $i \bowtie j$: Joining by an edge every vertex colored i to every vertex colored $j \neq i$
  - $\text{recolor}_{i,j}$: Recolor i into color j

- **Definition** Clique-Width of a graph G, denoted by $\text{cw}(G)$, is the minimum number of colors used for constructing G.

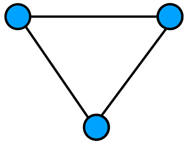
Example: Cliques

- NewVertex$_{blue}$

$n-1$ times

- NewVertex$_{red}$
- $blue \bowtie red$
- recolor$_{red,blue}$

- NewVertex$_{red}$
- $blue \bowtie red$
- recolor$_{red,blue}$

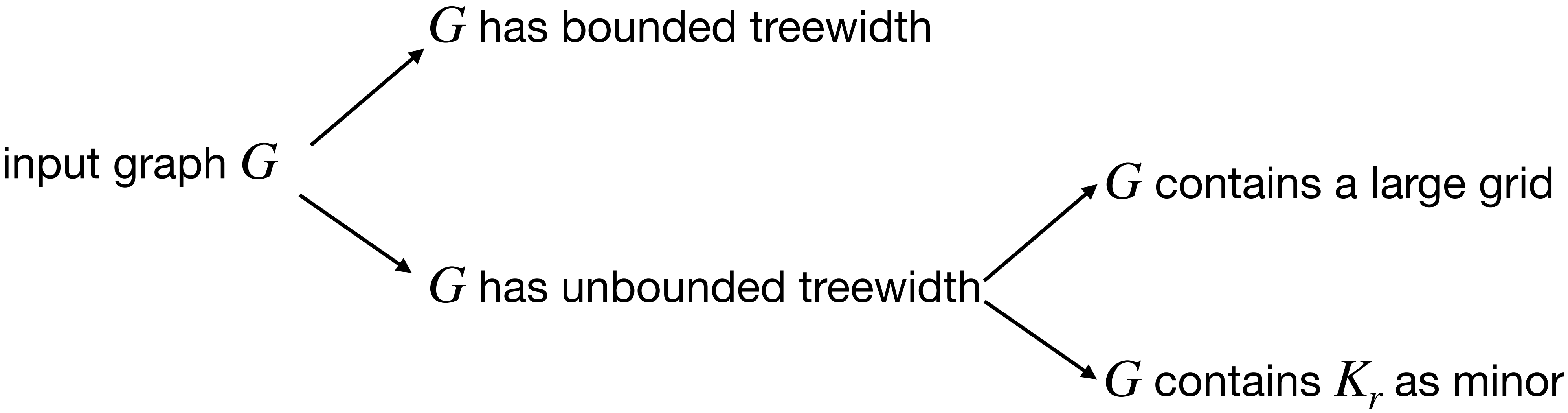


cw$(K_n) \leq 2$

Graph-Minor Theory

Theorem (Robertson, Seymour)
Every graph class closed under minor has a finite number of obstructions

Impact on algorithm design (parametrized complexity):



input graph G

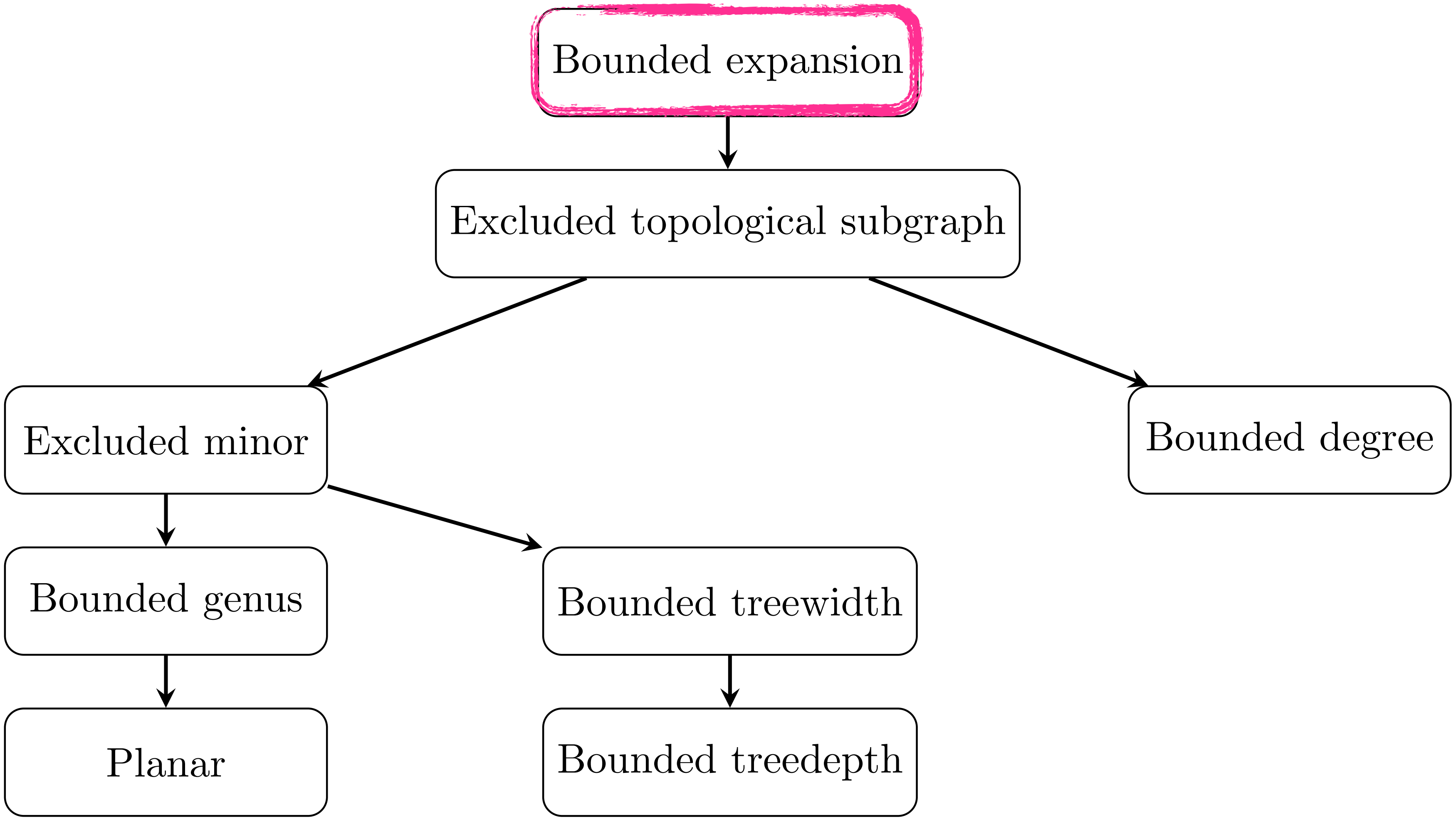
G has bounded treewidth

G has unbounded treewidth

G contains a large grid

G contains K_r as minor

Graph Classes



Bounded expansion

Excluded topological subgraph

Excluded minor

Bounded degree

Bounded genus

Bounded treewidth

Planar

Bounded treedepth

Classes of Bounded expansion

Definition A class $\mathscr{G}$ of graphs is of bounded expansion if there exists a function

$$f : \mathbb{N} \to \mathbb{N}$$

such that, for every $G \in \mathscr{G}$ and every $r \geq 0$, any graph H obtained from G by

1. contracting vertex-disjoint balls of radius at most r, and
2. deleting the remaining vertices

satisfies

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

Examples

Bounded degree graphs:

$$\mathcal{G}_\Delta = \{G = (V, E) \mid \forall v \in V, \deg(v) \leq \Delta\}$$

Expansion $f(r) = \Delta^{r+1}$

Planar graphs:

Expansion $f(r) = 6$

Other classes:

- graphs of bounded genus
- graphs of bounded treewidth
- graphs that exclude a fixed (topological) minor
- graphs of bounded stack number
- a.a.s., graphs drawn from $G_{n,p}$ with $p = O(1/n)$

Algorithmic Meta-Theorems

Theorem (Courcelle 1990)

Every graph property definable in the MSO_2 logic of graphs can be decided in linear time on graphs of bounded treewidth*.*

Theorem (Courcelle, Makowsky, Rotics, 2000)

Every graph property definable in the MSO_1 logic of graphs can be decided in linear time on graphs of bounded cliquewidth*.*

Theorem (Dvorák, Král, Thomas, 2010)

Every graph property definable in the FO logic of graphs can be decided in linear time on graph classes of bounded expansion*.*

Distributed
Algorithmic Meta-Theorems

Distributed Decision

Definition A distributed algorithm A decides a graph property Π if, for every graph $G = (V, E)$,

$$G \models \Pi \iff A \text{ accepts at all nodes.}$$

Can be extended to labeled graphs (G, ℓ) where $\ell : V(G) \to L$

- Is the graph properly 3-colored?
- Is the set of pointers forming a (minimum-weight) spanning tree?

FO on Bounded Expansion

Theorem (Blin, Fomin, F., Gay, Golovach, Montealegre, Rapaport, Todinca, 2025)

For every FO formula φ on graphs, and for every class of graphs $\mathcal{G}$ of bounded expansion, there exists a deterministic algorithm that, for every n-node network $G \in \mathcal{G}$ of diameter D, decides whether $G \models \varphi$ in $O(D + \log n)$ rounds in the CONGEST model.

Remark: $\Omega(D)$ rounds are necessary for formulas such as

$$\exists x \exists y \;\; \deg(x) = \deg(y) = 3$$

Local FO on Bounded Expansion

Theorem (Blin, Fomin, F., Gay, Golovach, Montealegre, Rapaport, Todinca, 2025)

For every local FO formula $\varphi(x)$ on graphs, and for every class of graphs $\mathcal{G}$ of bounded expansion, there exists a deterministic algorithm that, for every n-node network $G \in \mathcal{G}$, marks all vertices v of G satisfying $G \models \varphi(v)$, in $O(\log n)$ rounds in the CONGEST model.

Treedepth

Definition For any $d \geq 0$, a graph G has treedepth at most d if the n vertices of G can be embedded onto an n-node rooted forest F of depth at most d such that, for any edge of G, one of its end-points is mapped to an ancestor of the image of its other end-point in F.

Treedepth of G, $\text{td}(G)$, is the minimum d such that such a forest exists.

Lemma (Fomin, F., Montealegre, Rapaport, Todinca, 2024)
For every FO formula φ on graphs, and for every $d \geq 0$, there exists a deterministic algorithm that, for every n-node network G with $\text{td}(G) \leq d$, decides whether $G \models \varphi$, in $O(1)$ rounds in the CONGEST model.

Low Treedepth Coloring

Definition A graph class $\mathscr{G}$ admits a low treedepth coloring if there exists a function

$$f : \mathbb{N} \to \mathbb{N}$$

satisfying that, for every integer $p \geq 1$, the vertices of every graph $G \in \mathscr{G}$ can be properly colored using $f(p)$ colors such that, for every set

$$S \subseteq \{1, \ldots, f(p)\}$$

of at most p colors, the subgraph $G[S]$ of G induced by the vertices with colors in S satisfies $\mathrm{td}(G[S]) \leq |S|$.

Theorem (Nešetřil,Ossona de Mendez, 2008)

$\mathscr{G}$ has bounded expansion $\iff$ $\mathscr{G}$ admits a low treedepth coloring.

Remark $f(p) = O(\Delta^2 p)$ for $\mathscr{G}_\Delta$ and $f(p) = O(p \log p)$ for $\mathscr{G}_{\text{planar}}$

Efficient Distributed Algorithm for Low Treedepth Coloring

Theorem (Nešetřil,Ossona de Mendez, 2008)

Let $\mathcal{G}$ be a graph class of bounded expansion. There exists an $O(\log n)$-round algorithm in CONGEST that, for every given p, computes a treedepth coloring of any $G \in \mathcal{G}$ with $f(p)$ colors.

This may not be the 'best' f
not even the one describing $\mathcal{G}$

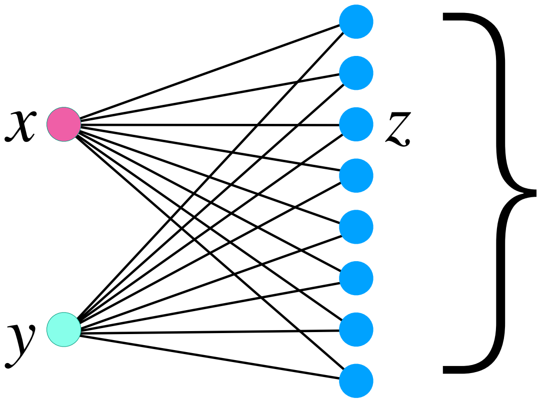
Application: Deciding H-Freeness

- Let $\mathscr{G}$ be a class of bounded expansion (e.g., planar graphs)
- Let $f : \mathbb{N} \to \mathbb{N}$ be such that $(p, f(p))$-treedepth coloring exists for every $p \geq 1$.
- Let $p = |V(H)|$
- Let $G \in \mathscr{G}$, and let $c : V(G) \to [f(p)]$ be a low-treedepth coloring.
- If G contains H as a subgraph, then there is a set S of at most p colors such that $G[S]$ contains H.
- For every $S \subseteq [f(p)]$ of p colors, testing whether $G[S]$ contains H is easy as $G[S]$ has treedepth at most p.

Algorithm: Try all $\binom{f(p)}{p}$ sets $S \subseteq [f(p)]$ of $p = |V(H)|$ colors.

Deciding Twins?

$$\mathsf{twin}(x) = \exists y \forall z \ \Big((\mathsf{adj}(x, z) \wedge \mathsf{adj}(y, z)) \vee (\neg \mathsf{adj}(x, z) \wedge \neg \mathsf{adj}(y, z)) \Big)$$



arbitrarily large number of common neighbors

Main Tool: Quantifier Elimination

$$\varphi(x) = Q_1 y_1 \; Q_2 y_2 \; \ldots \; Q_k y_k \; \psi(x, y_1, \ldots, y_k)$$

Goal: Constructing a *quantifier-free* formula $\widehat{\varphi}(x)$ such that, for every $G \in \mathscr{G}$

$$G \models \varphi(v) \iff \widehat{G} \models \widehat{\varphi}(v)$$

where $\widehat{G}$ is G enriched with additional unary predicates (i.e., labels).

Technique:

Bounded expansion

$\updownarrow$

Bounded treedepth

$\updownarrow$

Forests

Two Main Difficulties

1. Implementing all steps of quantifier-elimination in CONGEST, i.e.,
    (a) communications must remain local, and
    (b) communication are limited to $O(\log n)$ bits per round on each edges

2. Preserving the locality of the formula throughout the transformation

$$\begin{aligned}\text{twin}(x) &= \exists y \forall z \ \big((\text{adj}(x,z) \wedge \text{adj}(y,z)) \vee (\neg\text{adj}(x,z) \wedge \neg\text{adj}(y,z))\big) \\ &= \exists y \neg \exists z \ \big((\text{adj}(x,z) \wedge \neg\text{adj}(y,z)) \vee (\neg\text{adj}(x,z) \wedge \text{adj}(y,z))\big)\end{aligned}$$

This is addressed by adopting a redundant relativization of quantifiers.

Distributed Certification

Distributed Certification (1)

- Let Π be a Boolean predicate on (labeled) graphs
- **Certification scheme for Π**
  - **Prover =** centralized computationally-unlimited non-trustable oracle providing every node of the input labeled graph (G, ℓ) with a *certificate*
  - **Verifier =** 1-round distributed algorithm exchanging certificates between neighbors for checking whether the set of certificates is a distributed proof that $(G, \ell) \models \Pi$

Distributed Certification (2)

Completeness: $(G, \ell) \models \Pi \implies$ prover can give certificates such that verifier accepts at every node

Soundness: $(G, \ell) \not\models \Pi \implies$ for any set of certificates given by prover, verifier rejects in at least one node

Complexity measure = **size** of the certificates (for legal instances)

Examples

Theorem (Korman, Kutten, Peleg, 2005)
Certifying cycle-freeness can be done with certificates on $O(\log n)$ *bits.*

Theorem (Drucker, Kuhn, Oshman, 2014)
Certifying C_4*-freeness requires certificates on* $\tilde{\Omega}(\sqrt{n})$ *bits.*

Meta-Theorems for Certification

Theorem (Blin, Fomin, F., Gay, Golovach, Montealegre, Rapaport, Todinca, 2025)
For every FO formula φ, and for every graph class $\mathscr{G}$ of bounded ***expansion,*** *certifying $G \models \varphi$ can be implemented with certificates on $O(\log n)$ bits in n-node graphs $G \in \mathscr{G}$.*

Theorem (Cook, Kim, Masarik, 2025)
For every MSO_2 formula φ, certifying $G \models \varphi$ can be implemented with certificates on $O(\log n)$ bits in any n-node graph of bounded ***treewidth.***

Theorem (F., Montealegre, Todinca, Rapaport, 2022)
For every MSO_1 formula φ, certifying $G \models \varphi$ can be implemented with certificates on $O(\log^2 n)$ bits in any n-node graph of bounded ***cliquewidth.***

Conclusion

Open Problem

- Pushing FO Distributed Model Checking to nowhere dense graphs
- Compact distributed certification of H-minor-freeness
- Links beween sequential algorithms and CONGEST algorithms

⸮ poly time → polylog-round CONGEST → polylog space certification ?

Thank you!