Universität
Bremen

Advances in Algorithmic Meta-Theorems with Applications to Solution Discovery

Sebastian Siebertz

Parameterized Complexity of Computational Reasoning (PCCR 2026)

24.07.2026

Solution Discovery

Motivation

In many real-world settings, the current state of a system is already almost usable, but not yet perfect.

Goal: Find a new feasible solution by modifying the current one as cheaply as possible.

Solution Discovery

Motivation

In many real-world settings, the current state of a system is already almost usable, but not yet perfect.

Goal: Find a new feasible solution by modifying the current one as cheaply as possible.

Solution Discovery [FGMMRRSS 2023]

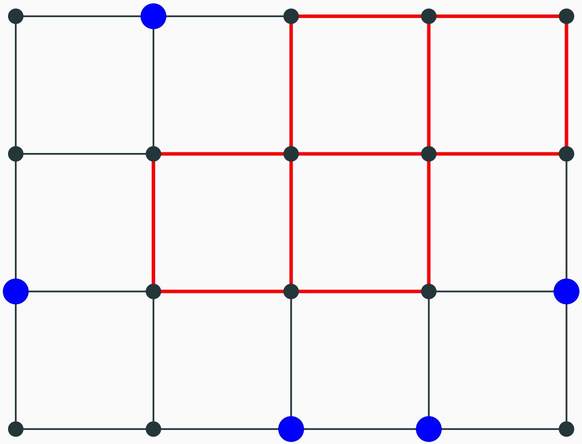
We model this as a graph problem inspired by combinatorial reconfiguration:

- we start with a given configuration (imagine tokens on vertices),
- we allow only small local changes (slides along edges),
- and we ask whether a valid solution can be reached within a given budget.

Feedback Vertex Set Discovery

Solution Discovery

- Input: A graph G, an initial set $S \subseteq V(G)$ of token positions, and a budget $b \in \mathbb{N}$.
- Question: Can we reach, by at most b token slides along edges, a set S' that forms a valid solution?

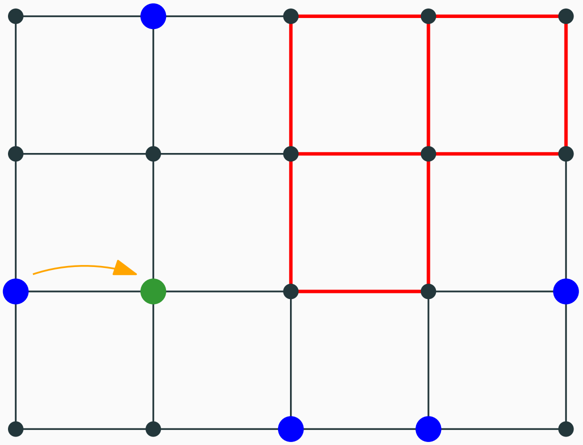


A discovery sequence for FEEDBACK VERTEX SET

Feedback Vertex Set Discovery

Solution Discovery

- Input: A graph G, an initial set $S \subseteq V(G)$ of token positions, and a budget $b \in \mathbb{N}$.
- Question: Can we reach, by at most b token slides along edges, a set S' that forms a valid solution?

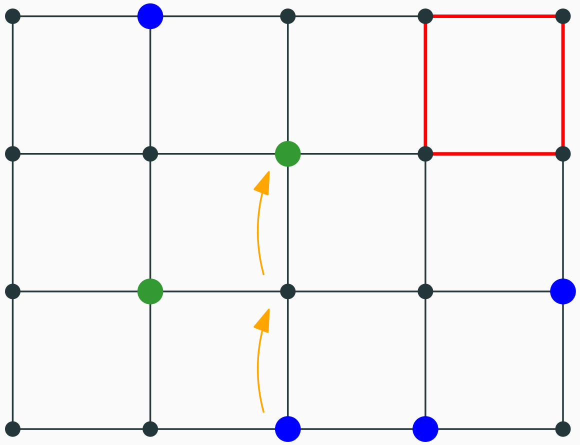


A discovery sequence for FEEDBACK VERTEX SET

Feedback Vertex Set Discovery

Solution Discovery

- Input: A graph G, an initial set $S \subseteq V(G)$ of token positions, and a budget $b \in \mathbb{N}$.
- Question: Can we reach, by at most b token slides along edges, a set S' that forms a valid solution?

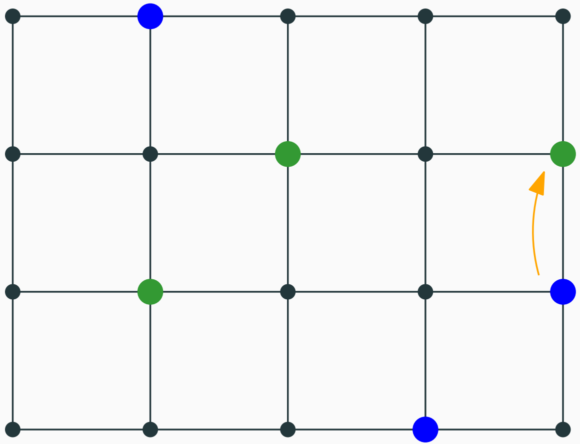


A discovery sequence for FEEDBACK VERTEX SET

Feedback Vertex Set Discovery

Solution Discovery

- Input: A graph G, an initial set $S \subseteq V(G)$ of token positions, and a budget $b \in \mathbb{N}$.
- Question: Can we reach, by at most b token slides along edges, a set S' that forms a valid solution?

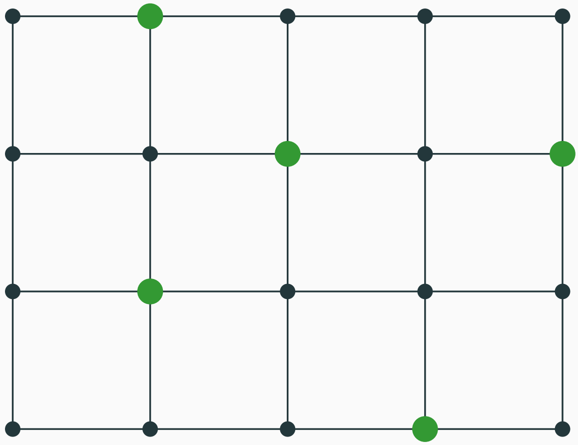


A discovery sequence for FEEDBACK VERTEX SET

Feedback Vertex Set Discovery

Solution Discovery

- Input: A graph G, an initial set $S \subseteq V(G)$ of token positions, and a budget $b \in \mathbb{N}$.
- Question: Can we reach, by at most b token slides along edges, a set S' that forms a valid solution?

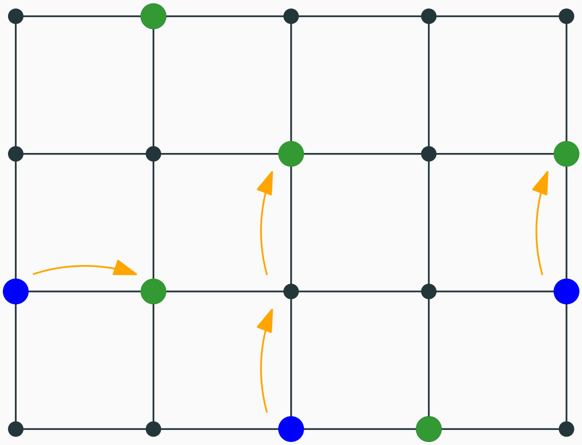


A discovery sequence for Feedback Vertex Set

Feedback Vertex Set Discovery

Solution Discovery

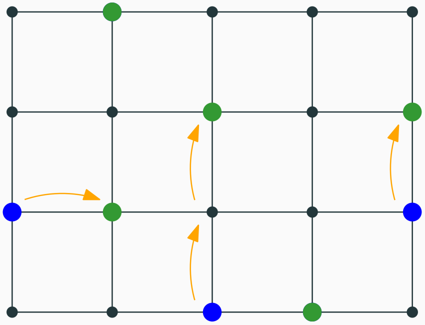
- Input: A graph G, an initial set $S \subseteq V(G)$ of token positions, and a budget $b \in \mathbb{N}$.
- Question: Can we reach, by at most b token slides along edges, a set S' that forms a valid solution?



A discovery sequence for FEEDBACK VERTEX SET

Feedback Vertex Set Discovery

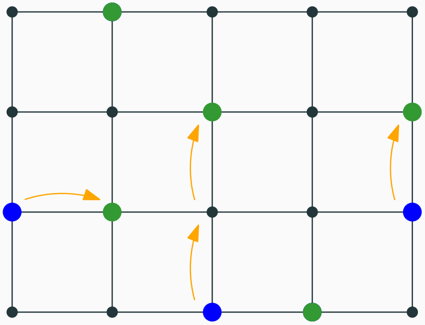
- Classically, FVS is NP-complete, and so is FVS-Discovery.



FVS-Discovery

Feedback Vertex Set Discovery

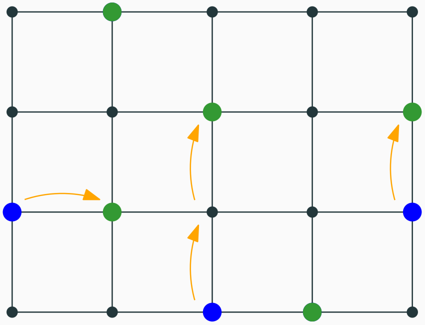
- Classically, FVS is NP-complete, and so is $\textsc{FVS-Discovery}$.
- Parameterized by the solution size k, $\textsc{FVS-Discovery}$ is FPT, that is, solvable in time $f(k) \cdot n^{\mathcal{O}(1)}$.



$\textsc{FVS-Discovery}$

Feedback Vertex Set Discovery

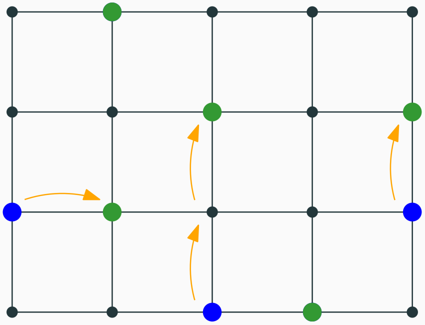
- Classically, FVS is NP-complete, and so is FVS-Discovery.
- Parameterized by the solution size k, FVS-Discovery is FPT, that is, solvable in time $f(k) \cdot n^{\mathcal{O}(1)}$.
- On general graphs, the problem is W[1]-hard for parameter b.



FVS-Discovery

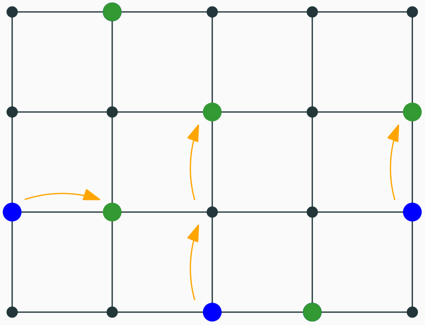
Feedback Vertex Set Discovery

- Classically, FVS is NP-complete, and so is FVS-Discovery.
- Parameterized by the solution size k, FVS-Discovery is FPT, that is, solvable in time $f(k) \cdot n^{\mathcal{O}(1)}$.
- On general graphs, the problem is W[1]-hard for parameter b.
- What are the most general graph classes on which the problem is tractable?



FVS-Discovery

- Classically, FVS is NP-complete, and so is $\mathrm{FVS\text{-}D{\scriptsize ISCOVERY}}$.
- Parameterized by the solution size k, $\mathrm{FVS\text{-}D{\scriptsize ISCOVERY}}$ is FPT, that is, solvable in time $f(k) \cdot n^{\mathcal{O}(1)}$.
- On general graphs, the problem is $\mathrm{W[1]}$-hard for parameter b.
- What are the most general graph classes on which the problem is tractable?



FVS-Discovery

Meta algorithmic approach

Can logic provide the right abstraction to reveal such tractability frontiers?

Logic as a tool to solve algorithmic problems

Model checking

Given a graph G and a formula φ: do we have $G \models \varphi$?

Logic as a tool to solve algorithmic problems

Model checking

Given a graph G and a formula φ: do we have $G \models \varphi$?

A formula for $\textsc{FVS-Discovery}$ (2 tokens move 2 steps each, a unary predicate T marks the initial token positions):

$$\varphi_{2,2} = \underbrace{\exists s_1, s_2, a_1, a_2, t_1, t_2 \left(T(s_1) \wedge T(s_2) \wedge \ E(s_1, a_1) \wedge E(a_1, t_1) \wedge \ E(s_2, a_2) \wedge E(a_2, t_2)\right.}_{\text{First-order logic (FO)}}$$

$$\left. \wedge \ \underbrace{\mathrm{Acyclic}(G - (T - \{s_1, s_2\} + \{t_1, t_2\}))}_{\text{Monadic second-order logic (MSO)}} \right).$$

Algorithmic meta-theorems

Courcelle's Theorem

Every problem definable in monadic second-order logic (MSO) can be solved in linear time on every graph class $\mathscr{C}$ with bounded treewidth/cliquewidth.

Algorithmic meta-theorems

Courcelle's Theorem

Every problem definable in monadic second-order logic (MSO) can be solved in linear time on every graph class $\mathscr{C}$ with bounded treewidth/cliquewidth.

$\Rightarrow$ FVS-DISCOVERY can be solved in time $f(b) \cdot n$ on classes with bounded treewidth/cliquewidth.

Algorithmic meta-theorems

Courcelle's Theorem

Every problem definable in monadic second-order logic (MSO) can be solved in linear time on every graph class $\mathscr{C}$ with bounded treewidth/cliquewidth.

$\Rightarrow$ $\textsc{FVS-Discovery}$ can be solved in time $f(b) \cdot n$ on classes with bounded treewidth/cliquewidth.

Let $\mathscr{C}$ be a weakly sparse class of graphs (excludes some biclique $K_{t,t}$ as a subgraph).

Algorithmic meta-theorems

Courcelle's Theorem

Every problem definable in monadic second-order logic (MSO) can be solved in linear time on every graph class $\mathscr{C}$ with bounded treewidth/cliquewidth.

$\Rightarrow$ FVS-Discovery can be solved in time $f(b) \cdot n$ on classes with bounded treewidth/cliquewidth.

Let $\mathscr{C}$ be a weakly sparse class of graphs (excludes some biclique $K_{t,t}$ as a subgraph).

- $\mathscr{C}$ has bounded treewidth $\Leftrightarrow$ MSO cannot transduce all graphs from $\mathscr{C}$:
  $\mathscr{C}$ is dependent for MSO

Algorithmic meta-theorems

Courcelle's Theorem

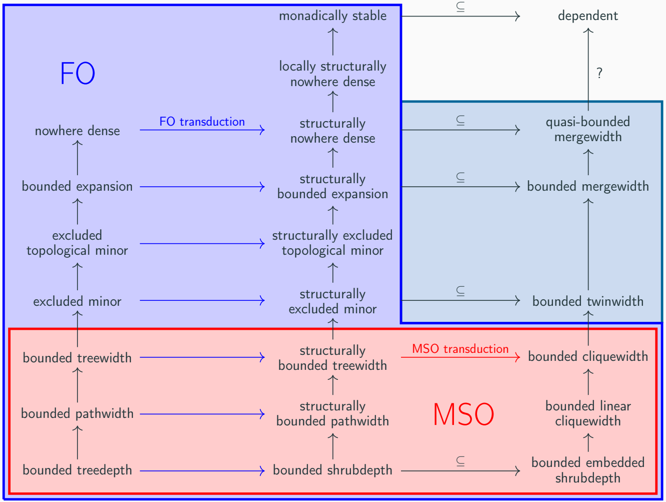
Every problem definable in monadic second-order logic (MSO) can be solved in linear time on every graph class $\mathscr{C}$ with bounded treewidth/cliquewidth.

$\Rightarrow$ FVS-Discovery can be solved in time $f(b) \cdot n$ on classes with bounded treewidth/cliquewidth.

Let $\mathscr{C}$ be a weakly sparse class of graphs (excludes some biclique $K_{t,t}$ as a subgraph).

- $\mathscr{C}$ has bounded treewidth $\Leftrightarrow$ MSO cannot transduce all graphs from $\mathscr{C}$:
  $\mathscr{C}$ is dependent for MSO
- Courcelle's Theorem captures the limit of tractability for MSO.
  (under a mild condition on efficiency of encoding)

Tractable graph classes



Back to discovery: what can plain FO express?

- The movement of at most b tokens can be expressed in FO:

Back to discovery: what can plain FO express?

- The movement of at most b tokens can be expressed in FO:
  - we existentially quantify the start and target vertices,

Back to discovery: what can plain FO express?

- The movement of at most b tokens can be expressed in FO:
  - we existentially quantify the start and target vertices,
  - we describe the slides such that the number of steps is at most b.

Back to discovery: what can plain FO express?

- The movement of at most b tokens can be expressed in FO:
  - we existentially quantify the start and target vertices,
  - we describe the slides such that the number of steps is at most b.
- But acyclicity of the target configuration cannot be expressed in FO.

Back to discovery: what can plain FO express?

- The movement of at most b tokens can be expressed in FO:
  - we existentially quantify the start and target vertices,
  - we describe the slides such that the number of steps is at most b.
- But acyclicity of the target configuration cannot be expressed in FO.

Bottleneck

To formulate FVS-discovery, we need to talk about the disappearance of cycles.

Back to discovery: what can plain FO express?

- The movement of at most b tokens can be expressed in FO:
  - we existentially quantify the start and target vertices,
  - we describe the slides such that the number of steps is at most b.
- But acyclicity of the target configuration cannot be expressed in FO.

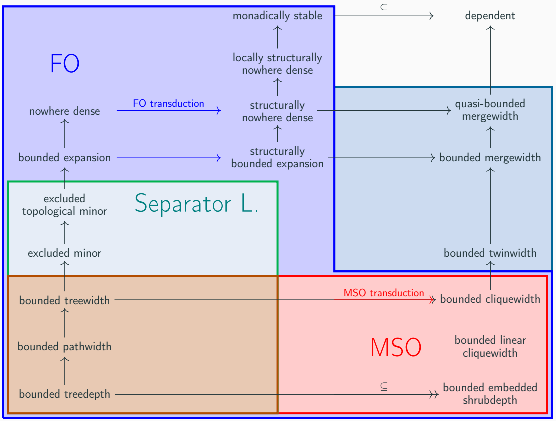
Bottleneck

To formulate FVS-discovery, we need to talk about the disappearance of cycles.

- MSO can formulate the problem, but then we have tractability only on classes with bounded treewidth/cliquewidth.

Separator logic

- FO is too weak for connectivity and acyclicity.

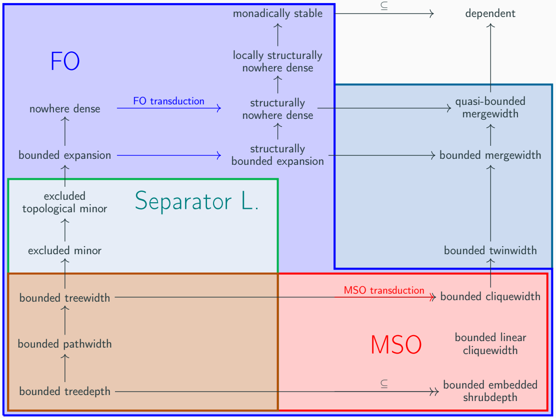


Separator logic

- FO is too weak for connectivity and acyclicity.
- Separator logic extends FO by predicates

$$\mathrm{conn}(x, y; \bar{z}),$$

that is, connectivity after deleting finitely many vertices [B 2022, SSV 2022].



Separator logic

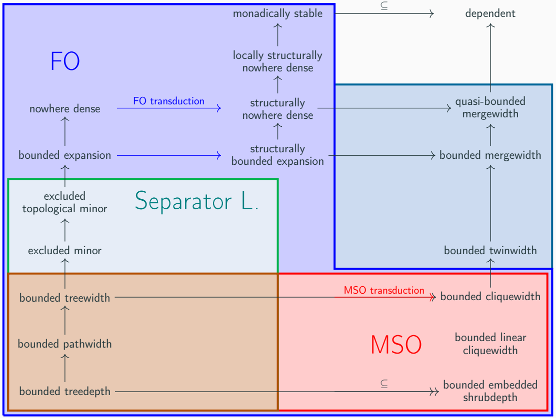
- FO is too weak for connectivity and acyclicity.
- Separator logic extends FO by predicates

$$\mathrm{conn}(x, y; \bar{z}),$$

  that is, connectivity after deleting finitely many vertices [B 2022, SSV 2022].
- On weakly sparse classes:

  $\mathscr{C}$ dependent for separator logic $\Leftrightarrow$

  $\mathscr{C}$ excludes a topological minor and model checking is efficient on $\mathscr{C}$ [PSSTV 2022].



Subconnectivity logic

- Separator logic talks about connectivity in the whole graph, but an unbounded number of vertices does not have to be considered, as they are occupied by tokens.

Subconnectivity logic

- Separator logic talks about connectivity in the whole graph, but an unbounded number of vertices does not have to be considered, as they are occupied by tokens.
- For FVS-Discovery we need to reason only inside the token-free subgraph.

Subconnectivity logic

- Separator logic talks about connectivity in the whole graph, but an unbounded number of vertices does not have to be considered, as they are occupied by tokens.
- For FVS-Discovery we need to reason only inside the token-free subgraph.
- Let G be a graph and let $Z \subseteq E(G)$ be a set of marked edges. Let $G_Z = (V(G), Z)$.

Subconnectivity logic

- Separator logic talks about connectivity in the whole graph, but an unbounded number of vertices does not have to be considered, as they are occupied by tokens.
- For FVS-Discovery we need to reason only inside the token-free subgraph.
- Let G be a graph and let $Z \subseteq E(G)$ be a set of marked edges. Let $G_Z = (V(G), Z)$.

Subconnectivity logic [MOdMSSSTV 2026+]

FO+subconn extends FO by atoms

$$\mathrm{subconn}(x, y; \bar{z}),$$

with the meaning: x and y are connected in G_Z after deleting $\bar{z}$.

Subconnectivity logic

Theorem [MOdMSSSTV 2026+]

Let $\mathscr{C}$ be a weakly sparse class of graphs.

Theorem [MOdMSSSTV 2026+]

Let $\mathscr{C}$ be a weakly sparse class of graphs.

- $\mathscr{C}$ excludes a minor $\Leftrightarrow$ $\mathscr{C}$ is dependent for Subconnectivity logic.

Subconnectivity logic

Theorem [MOdMSSSTV 2026+]

Let $\mathscr{C}$ be a weakly sparse class of graphs.

- $\mathscr{C}$ excludes a minor $\Leftrightarrow$ $\mathscr{C}$ is dependent for Subconnectivity logic.
- $\mathscr{C}$ excludes a minor $\Rightarrow$ model checking for Subconnectivity logic is tractable on $\mathscr{C}$.

Subconnectivity logic

Theorem [MOdMSSSTV 2026+]

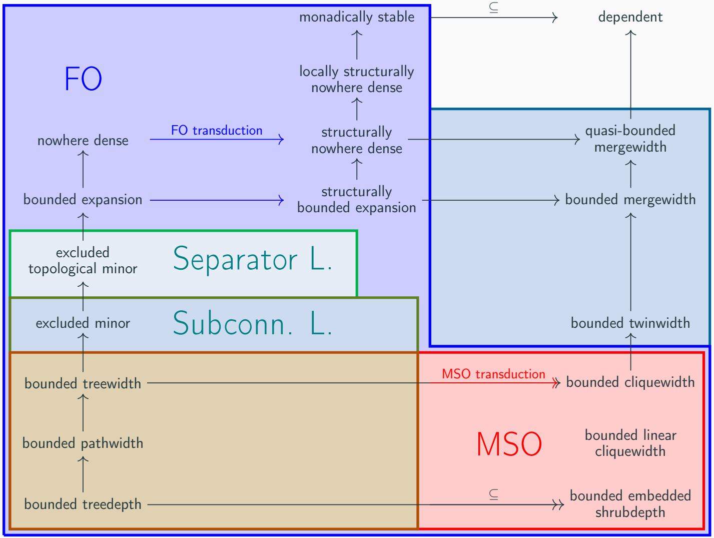
Let $\mathscr{C}$ be a weakly sparse class of graphs.

- $\mathscr{C}$ excludes a minor $\Leftrightarrow$ $\mathscr{C}$ is dependent for Subconnectivity logic.
- $\mathscr{C}$ excludes a minor $\Rightarrow$ model checking for Subconnectivity logic is tractable on $\mathscr{C}$.

Corollary

$\mathrm{FVS\text{-}D{\scriptsize ISCOVERY}}$ on classes with excluded minors is FPT with respect to parameter b.

Subconnectivity Logic



Proof sketch

Key observation

Let Y be a depth-first search spanning tree of a (connected) graph G and let $\preceq_Y$ be the corresponding tree-order.

Then there is a formula $\psi \in \mathrm{FO}[E, \preceq]$ such that

$$G \models \mathrm{conn}(x, y, \bar{z}) \Leftrightarrow (G, \preceq_Y) \models \psi(x, y, \bar{z}).$$

Proof of the observation

Tree-ordered graph classes

- A weakly sparse hereditary class of tree-ordered graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ its tree-ordered minors are nowhere dense.

Tree-ordered graph classes

- A weakly sparse hereditary class of tree-ordered graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ its tree-ordered minors are nowhere dense.

  A weakly sparse hereditary class of <u>spanning tree-ordered</u> graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ $\mathscr{C}$ excludes a minor.

  [BLNOdMS, 2026+]

Tree-ordered graph classes

- A weakly sparse hereditary class of tree-ordered graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ its tree-ordered minors are nowhere dense.

  A weakly sparse hereditary class of <u>spanning tree-ordered</u> graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ $\mathscr{C}$ excludes a minor.

  [BLNOdMS, 2026+]
- A <u>linearly ordered</u> hereditary class of graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ $\mathscr{C}$ has bounded twinwidth.

Tree-ordered graph classes

- A weakly sparse hereditary class of tree-ordered graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ its tree-ordered minors are nowhere dense.

  A weakly sparse hereditary class of spanning tree-ordered graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ $\mathscr{C}$ excludes a minor.

  [BLNOdMS, 2026+]

- A linearly ordered hereditary class of graphs $\mathscr{C}$ is dependent for FO $\Leftrightarrow$ $\mathscr{C}$ has bounded twinwidth.

  Furthermore, FO model checking is FPT on such classes.
  [Bonnet et al. 2024]

Tree-orders vs linear orders

- Let $\mathscr{C}$ be a class with bounded twinwidth. Then every FO-definable expansion of $\mathscr{C}$ has bounded twinwidth [Bonnet et. al 2021].

Tree-orders vs linear orders

- Let $\mathscr{C}$ be a class with bounded twinwidth. Then every FO-definable expansion of $\mathscr{C}$ has bounded twinwidth [Bonnet et. al 2021].
- If the graphs in $\mathscr{C}$ have bounded degree, then from a spanning tree order we can transduce a linear order [BLNOdMS, 2026+].

Tree-orders vs linear orders

- Let $\mathscr{C}$ be a class with bounded twinwidth. Then every FO-definable expansion of $\mathscr{C}$ has bounded twinwidth [Bonnet et. al 2021].
- If the graphs in $\mathscr{C}$ have bounded degree, then from a spanning tree order we can transduce a linear order [BLNOdMS, 2026+].

Question: how to deal with high degree vertices in spanning trees in a definable way?

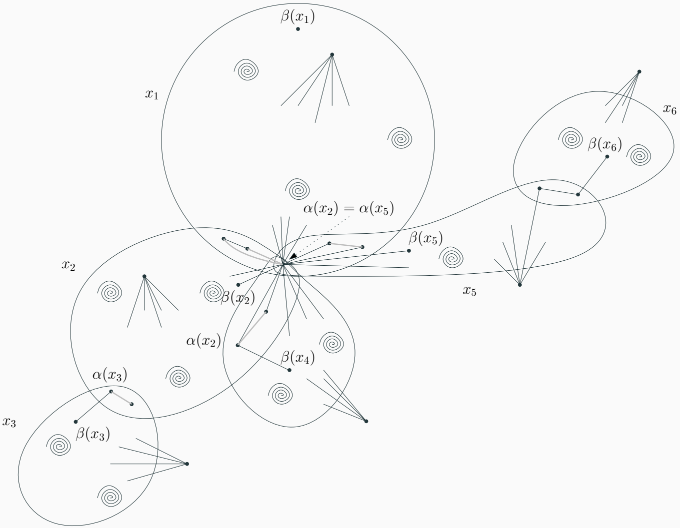
Tree-orders vs linear orders

- Let $\mathscr{C}$ be a class with bounded twinwidth. Then every FO-definable expansion of $\mathscr{C}$ has bounded twinwidth [Bonnet et. al 2021].
- If the graphs in $\mathscr{C}$ have bounded degree, then from a spanning tree order we can transduce a linear order [BLNOdMS, 2026+].

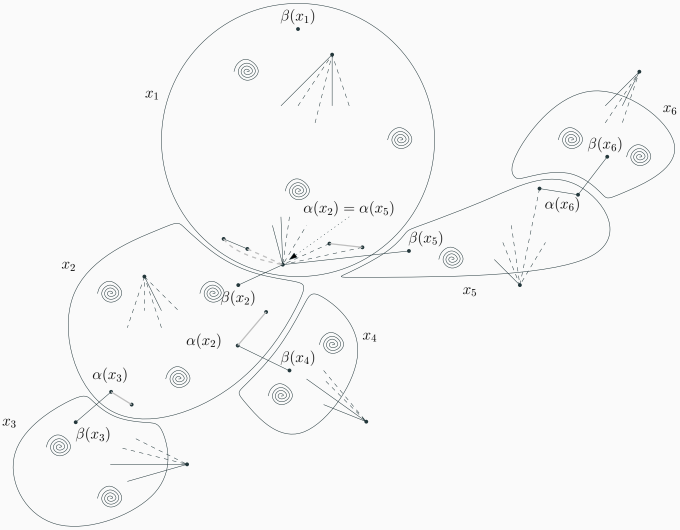
Question: how to deal with high degree vertices in spanning trees in a definable way?

- If $\mathscr{C}$ is apex minor free, then there exists an apex minor free class $\mathscr{D}$ of subcubic graphs such that for every $G \in \mathscr{C}$ there exists $H \in \mathscr{D}$ with $G \leq_m H$. [Milolidakis, 2025]

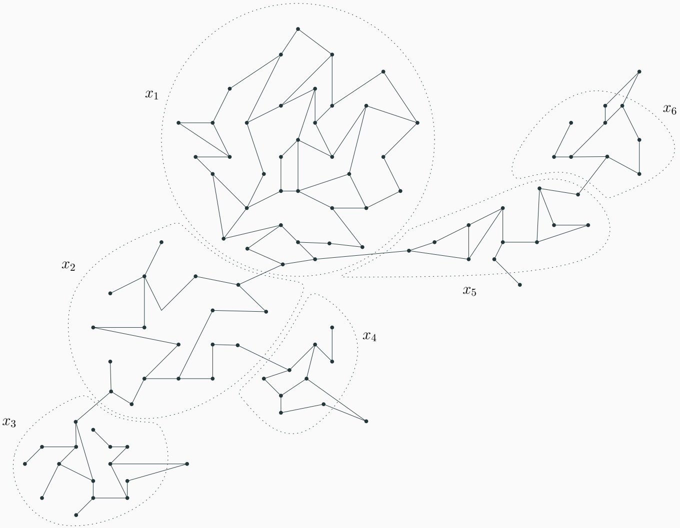
The structure theorem of Robertson and Seymour



Killing the apices



Apex minor free graphs [Milolidakis, 2025]



Summary

- Reduce Subconnectivity logic to FO with depth-first search spanning tree orders.

Summary

- Reduce Subconnectivity logic to FO with depth-first search spanning tree orders.
- Given a graph $G \in \mathscr{C}$, compute a subcubic graph H excluding a fixed apex minor and a spanning tree Y_H of H such that G and a dfs tree Y_G of G is definable in H.

Summary

- Reduce Subconnectivity logic to FO with depth-first search spanning tree orders.
- Given a graph $G \in \mathscr{C}$, compute a subcubic graph H excluding a fixed apex minor and a spanning tree Y_H of H such that G and a dfs tree Y_G of G is definable in H.
- The class of all (H, Y_H) can be linearly ordered in a definable way and hence has bounded twinwidth and model checking is tractable.

Summary

- Reduce Subconnectivity logic to FO with depth-first search spanning tree orders.
- Given a graph $G \in \mathscr{C}$, compute a subcubic graph H excluding a fixed apex minor and a spanning tree Y_H of H such that G and a dfs tree Y_G of G is definable in H.
- The class of all (H, Y_H) can be linearly ordered in a definable way and hence has bounded twinwidth and model checking is tractable.

Corollary [MOdMSSTV 2026+]

Let $\mathscr{C}$ be a weakly sparse hereditary class of graphs. The following are equivalent.

1. $\mathscr{C}$ excludes a minor.
2. The expansion of $\mathscr{C}$ by all spanning forest orders is dependent.
3. The expansion of $\mathscr{C}$ by all spanning forest orders has bounded twinwidth.

Subconnectivity logic

Theorem [MOdMSSTV 2026+]

Model checking for FO+subconn is FPT on classes with excluded minors.

Subconnectivity logic

Theorem [MOdMSSTV 2026+]

Model checking for FO+subconn is FPT on classes with excluded minors.

Meta-theorem for solution discovery

For every fixed FO+subconn-formula $\psi(T)$, ψ-Modulator Discovery on classes with an excluded minor is fixed-parameter tractable with respect to b.

Theorem [MOdMSSTV 2026+]

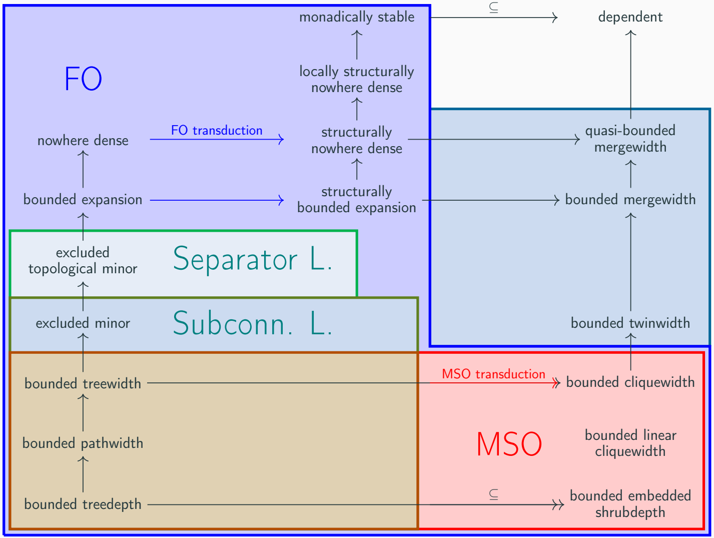
Model checking for FO+subconn is FPT on classes with excluded minors.

Meta-theorem for solution discovery

For every fixed FO+subconn-formula $\psi(T)$, ψ-$\textsc{Modulator Discovery}$ on classes with an excluded minor is fixed-parameter tractable with respect to b.

- In particular, $\textsc{FVS-Discovery}$ is FPT on classes with an excluded minor, parameterized by b.

Subconnectivity Logic



Literature

- **FGMMRRSS 2023**: Michael R. Fellows, Mario Grobler, Nicole Megow, Amer E. Mouawad, Vijayaragunathan Ramamoorthi, Frances A. Rosamond, Daniel Schmand, and Sebastian Siebertz. *On Solution Discovery via Reconfiguration*.
- **B 2022**: Mikołaj Bojańczyk. *Separator Logic and Star-Free Expressions for Graphs*.
- **SSV 2022**: Nicole Schirrmacher, Sebastian Siebertz, and Alexandre Vigny. *First-Order Logic with Connectivity Operators*.
- **PSSTV 2022**: Michał Pilipczuk, Nicole Schirrmacher, Sebastian Siebertz, Szymon Toruńczyk, and Alexandre Vigny. *Algorithms and Data Structures for First-Order Logic with Connectivity Under Vertex Failures*.
- **MOdMSSSTV 2026+**: Nikolas Mählmann, Patrice Ossona de Mendez, Nicole Schirrmacher, Sebastian Siebertz, Giannos Stamoulis, Dimitrios M. Thilikos, and Alexandre Vigny. *A Logic for Minor-Free Graph Classes: Model Checking, Dependence, and Combinatorial Reconfiguration. Unpublished.*.
- **BLNOdMS 2026+**: Hector Buffière, Yuquan Lin, Jaroslav Nešetřil, Patrice Ossona de Mendez, and Sebastian Siebertz. *Characterizations of Monadically Dependent Tree-Ordered Weakly Sparse Structures*.
- **Bonnet et al. 2024**: Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Toruńczyk. *Twin-width IV: Ordered Graphs and Matrices*.
- **Bonnet et al. 2021**: Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. *Twin-width I: Tractable FO Model Checking*.
- **Milolidakis 2025**: Orestis C. Milolidakis. *Splittability within Minor-Closed Classes to Graphs of Low Maximum Degree*. Thesis.