

Algorithmic and Finite Model Theory

Martin Grohe

Department of Computer Science
Humboldt University Berlin



Algorithmic and Finite Model Theory
on Restricted Classes of Structures

1. Directions and Results
2. Classes of Graphs
3. Techniques from Logic
4. An Algorithmic Meta Theorem

1. Directions and Results
2. Classes of Graphs
3. Techniques from Logic
4. An Algorithmic Meta Theorem

“Classical” Model Theoretic Questions

Background

Fundamental theorems of classical model theory fail when restricted to finite structures:

Compactness theorem, extension preservation theorem, interpolation theorem

(Gurevich 1984)

“Classical” Model Theoretic Questions

Background

Fundamental theorems of classical model theory fail when restricted to finite structures:

Compactness theorem, extension preservation theorem, interpolation theorem

(Gurevich 1984)

Exception (Rossman 2005)

The **homomorphism preservation theorem** holds in the finite.

“Classical” Model Theoretic Questions (cont’d)

Theorem (Atserias, Dawar, Kolaitis 2004)

The homomorphism preservation theorem holds when restricted to various classes of finite graphs such as

- ▶ *planar graphs,*
- ▶ *graphs of bounded tree width,*
- ▶ *classes defined by excluded minors,*
- ▶ *graphs of bounded degree.*

“Classical” Model Theoretic Questions (cont’d)

Theorem (Atserias, Dawar, Kolaitis 2004)

The homomorphism preservation theorem holds when restricted to various classes of finite graphs such as

- ▶ *planar graphs,*
- ▶ *graphs of bounded tree width,*
- ▶ *classes defined by excluded minors,*
- ▶ *graphs of bounded degree.*

Theorem (Atserias, Dawar, G. 2005)

The extension preservation theorem holds when restricted to classes of finite graphs of bounded tree width or bounded degree

“Classical” Model Theoretic Questions (cont’d)

Theorem (Atserias, Dawar, Kolaitis 2004)

The homomorphism preservation theorem holds when restricted to various classes of finite graphs such as

- ▶ *planar graphs,*
- ▶ *graphs of bounded tree width,*
- ▶ *classes defined by excluded minors,*
- ▶ *graphs of bounded degree.*

Theorem (Atserias, Dawar, G. 2005)

The extension preservation theorem holds when restricted to classes of finite graphs of bounded tree width or bounded degree

(but not the class of planar graphs).

“Classical” Model Theoretic Questions (cont’d)

FO = first-order logic

Theorem (Benedikt, Segoufin 2004)

FO = order-invariant FO(\leq) on finite trees.

(weak form of interpolation).

“Classical” Model Theoretic Questions (cont’d)

FO = first-order logic

Theorem (Benedikt, Segoufin 2004)

FO = order-invariant FO(\leq) on finite trees.

(weak form of interpolation).

ML = modal logic

Theorem (Dawar, Otto 2005)

ML = bisimulation invariant FO on finite reflexive and transitive Kripke structures

Theorem (Folklore)

The following logics have the same expressive power on the class of finite trees:

1. *monadic second-order logic*
2. *monadic least fixed-point logic*
3. *monadic datalog*
4. *modal μ -calculus*

(but they differ in succinctness).

Expressiveness (cont'd)

Theorem (G., Mariño 1999)

Monadic second order logic is strictly less expressive than least fixed-point logic on classes of structures of bounded tree width.

MSO = monadic second-order logic

MSO = monadic second-order logic

GSO = guarded second order logic
(allows quantification over subsets of the relations in
a structure, e.g. sets of edges in a graph)

MSO = monadic second-order logic

GSO = guarded second order logic
(allows quantification over subsets of the relations in
a structure, e.g. sets of edges in a graph)

Theorem (Seese 1991)

1. *For every $k \geq 1$, the GSO-theory of the class \mathcal{T}_k of graphs of tree width $\leq k$ is decidable.*

MSO = monadic second-order logic

GSO = guarded second order logic
(allows quantification over subsets of the relations in
a structure, e.g. sets of edges in a graph)

Theorem (Seese 1991)

1. *For every $k \geq 1$, the GSO-theory of the class \mathcal{T}_k of graphs of tree width $\leq k$ is decidable.*
2. *If a class \mathcal{C} of graphs has a decidable GSO-theory, then there is a $k \geq 1$ such that $\mathcal{C} \subseteq \mathcal{T}_k$.*

Theorem (Seese, Courcelle, ...)

For every $k \geq 1$, the MSO-theory of the class \mathcal{R}_k of graphs of rank width $\leq k$ is decidable.

Theorem (Seese, Courcelle, ...)

For every $k \geq 1$, the MSO-theory of the class \mathcal{R}_k of graphs of rank width $\leq k$ is decidable.

Seese's Conjecture

If a class \mathcal{C} of graphs has a decidable MSO-theory, then there is a $k \geq 1$ such that $\mathcal{C} \subseteq \mathcal{R}_k$.

Algorithmic Meta Theorems

Theorem (Courcelle 1990)

GSO-definable properties of graphs of bounded tree width are decidable in linear time.

Algorithmic Meta Theorems

Theorem (Courcelle 1990)

GSO-definable properties of graphs of bounded tree width are decidable in linear time.

Theorem (Seese 1996)

FO-definable properties of graphs of bounded degree are decidable in linear time.

Algorithmic Meta Theorems

Theorem (Courcelle 1990)

GSO-definable properties of graphs of bounded tree width are decidable in linear time.

Theorem (Seese 1996)

FO-definable properties of graphs of bounded degree are decidable in linear time.

Theorem (Frick, G. 2001)

FO-definable properties of planar graphs are decidable in linear time.

Algorithmic Meta Theorems (cont'd)

Theorem (Flum, Frick, G. 2002)

GSO-definable queries on graphs of bounded tree width can be evaluated in time linear in the input size plus output size.

Algorithmic Meta Theorems (cont'd)

Theorem (Flum, Frick, G. 2002)

GSO-definable queries on graphs of bounded tree width can be evaluated in time linear in the input size plus output size.

Theorem (Dawar, G., Kreutzer, Schweikardt 2006)

FO-definable optimisation problem on graphs with excluded minors have polynomial time approximation schemes.

Descriptive Complexity

IFP+C = Inflationary fixed-point logic with counting

Descriptive Complexity

IFP+C = Inflationary fixed-point logic with counting

Theorem (G., Mariño 1999)

IFP+C captures polynomial time on classes of graphs of bounded tree width.

IFP+C = Inflationary fixed-point logic with counting

Theorem (G., Mariño 1999)

IFP+C captures polynomial time on classes of graphs of bounded tree width.

That is, for every property \mathcal{P} of graphs of bounded tree width:

\mathcal{P} definable in IFP+C \iff \mathcal{P} decidable in polynomial time.

IFP+C = Inflationary fixed-point logic with counting

Theorem (G., Mariño 1999)

IFP+C captures polynomial time on classes of graphs of bounded tree width.

That is, for every property \mathcal{P} of graphs of bounded tree width:

\mathcal{P} definable in IFP+C \iff \mathcal{P} decidable in polynomial time.

Theorem (G. 2008)

IFP+C captures polynomial time on the class of all graphs with no K_5 -minor.

Identifying Graphs

FO+C = First-order logic with counting

FO+C = First-order logic with counting

Theorem (Verbitzky, G. 2006, Verbitzky 2007)

1. *For every n -vertex planar graph G there is an FO+C-sentence of quantifier rank $O(\log n)$ that distinguishes G from all other graphs.*

FO+C = First-order logic with counting

Theorem (Verbitzky, G. 2006, Verbitzky 2007)

1. *For every n -vertex planar graph G there is an FO+C-sentence of quantifier rank $O(\log n)$ that distinguishes G from all other graphs.*
2. *For every n -vertex graph G of tree width k there is an FO+C-sentence with $O(k)$ variables and quantifier rank $O(\log n)$ that distinguishes G from all other graphs.*

FO+C = First-order logic with counting

Theorem (Verbitzky, G. 2006, Verbitzky 2007)

1. *For every n -vertex planar graph G there is an FO+C-sentence of quantifier rank $O(\log n)$ that distinguishes G from all other graphs.*
2. *For every n -vertex graph G of tree width k there is an FO+C-sentence with $O(k)$ variables and quantifier rank $O(\log n)$ that distinguishes G from all other graphs.*

Corollary

Isomorphism testing for graphs of bounded tree width is in NC (more precisely: TC^1).

Algorithmic Learning Theory

A structure,

$\varphi(\vec{x}, \vec{y})$ formula where $\vec{x} = (x_1, \dots, x_k), \vec{y} = (y_1, \dots, y_\ell)$

For $\vec{b} \in A^\ell$:

$$\varphi(A, \vec{b}) := \{\vec{a} \in A^k \mid A \models \varphi(\vec{a}, \vec{b})\}$$

Algorithmic Learning Theory

A structure,

$\varphi(\vec{x}, \vec{y})$ formula where $\vec{x} = (x_1, \dots, x_k), \vec{y} = (y_1, \dots, y_\ell)$

For $\vec{b} \in A^\ell$:

$$\varphi(A, \vec{b}) := \{\vec{a} \in A^k \mid A \models \varphi(\vec{a}, \vec{b})\}$$

Concept class defined by φ on A :

$$\mathcal{C}(A, \varphi) := \{\varphi(A, \vec{b}) \mid \vec{b} \in A^\ell\}.$$

(family of relations).

Algorithmic Learning Theory

A structure,

$\varphi(\vec{x}, \vec{y})$ formula where $\vec{x} = (x_1, \dots, x_k), \vec{y} = (y_1, \dots, y_\ell)$

For $\vec{b} \in A^\ell$:

$$\varphi(A, \vec{b}) := \{\vec{a} \in A^k \mid A \models \varphi(\vec{a}, \vec{b})\}$$

Concept class defined by φ on A :

$$\mathcal{C}(A, \varphi) := \{\varphi(A, \vec{b}) \mid \vec{b} \in A^\ell\}.$$

(family of relations).

Example

$A := (\mathbb{R}, +, \cdot, 0, 1, \leq)$ ordered field of reals

$$\varphi(x_1, x_2, y_1, y_2, y_3) := ((x_1 - y_1) \cdot (x_1 - y_1) + (x_2 - y_2) \cdot (x_2 - y_2) \leq y_3 \cdot y_3).$$

Algorithmic Learning Theory

A structure,

$\varphi(\vec{x}, \vec{y})$ formula where $\vec{x} = (x_1, \dots, x_k), \vec{y} = (y_1, \dots, y_\ell)$

For $\vec{b} \in A^\ell$:

$$\varphi(A, \vec{b}) := \{\vec{a} \in A^k \mid A \models \varphi(\vec{a}, \vec{b})\}$$

Concept class defined by φ on A :

$$\mathcal{C}(A, \varphi) := \{\varphi(A, \vec{b}) \mid \vec{b} \in A^\ell\}.$$

(family of relations).

Example

$A := (\mathbb{R}, +, \cdot, 0, 1, \leq)$ ordered field of reals

$$\varphi(x_1, x_2, y_1, y_2, y_3) := ((x_1 - y_1) \cdot (x_1 - y_1) + (x_2 - y_2) \cdot (x_2 - y_2) \leq y_3 \cdot y_3).$$

Then $\mathcal{C}(A, \varphi)$ is the class of disks in the plane.

Algorithmic Learning Theory (cont'd)

Theorem (G., Turan 2004)

1. *GSO-definable concept classes on graphs of bounded tree width are learnable in the PAC-model.*
2. *FO-definable concept classes on graphs of bounded local rank width are learnable in the PAC-model.*

Constraint Satisfaction Problems

Graph of a CSP-instance:

- ▶ vertices are variables
- ▶ edges are between variables that appear together in a constraint

Constraint Satisfaction Problems

Graph of a CSP-instance:

- ▶ vertices are variables
- ▶ edges are between variables that appear together in a constraint

For class \mathcal{C} of graphs:

$\text{CSP}(\mathcal{C}) := \text{CSP}$ on instances with graph in \mathcal{C}

Constraint Satisfaction Problems (cont'd)

Theorem (G., Schwentick, Segoufin 2001)

For every class \mathcal{C} of graphs:

$\text{CSP}(\mathcal{C})$ in polynomial time $\iff \mathcal{C}$ has bounded tree width

(under a reasonable complexity theoretic assumption).

Constraint Satisfaction Problems (cont'd)

Theorem (G., Schwentick, Segoufin 2001)

For every class \mathcal{C} of graphs:

$\text{CSP}(\mathcal{C})$ in polynomial time $\iff \mathcal{C}$ has bounded tree width

(under a reasonable complexity theoretic assumption).

Theorem (Marx 2007)

For every class \mathcal{C} of graphs of tree width k

$\text{CSP}(\mathcal{C})$ solvable in time $n^{O(k)}$ but not in time $n^{O(k/\log k)}$.

(under a reasonable complexity theoretic assumption).

1. Directions and Results

2. Classes of Graphs

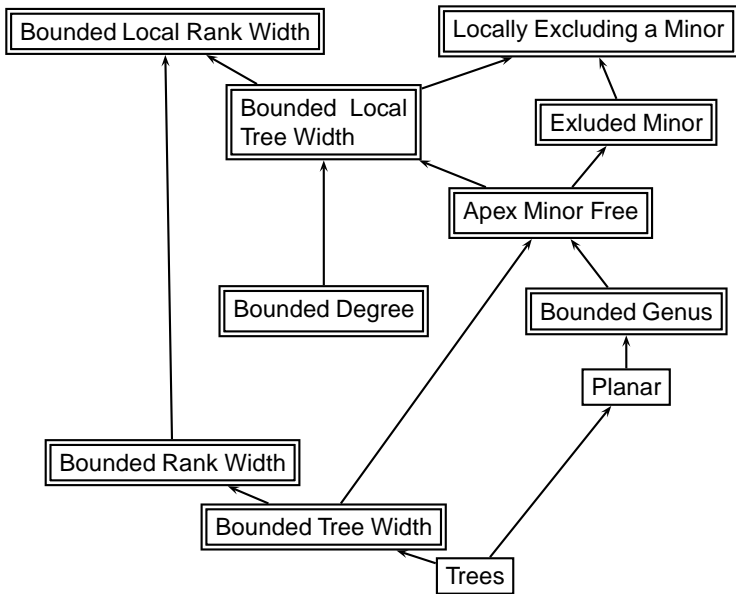
Tree Decompositions and Tree Width

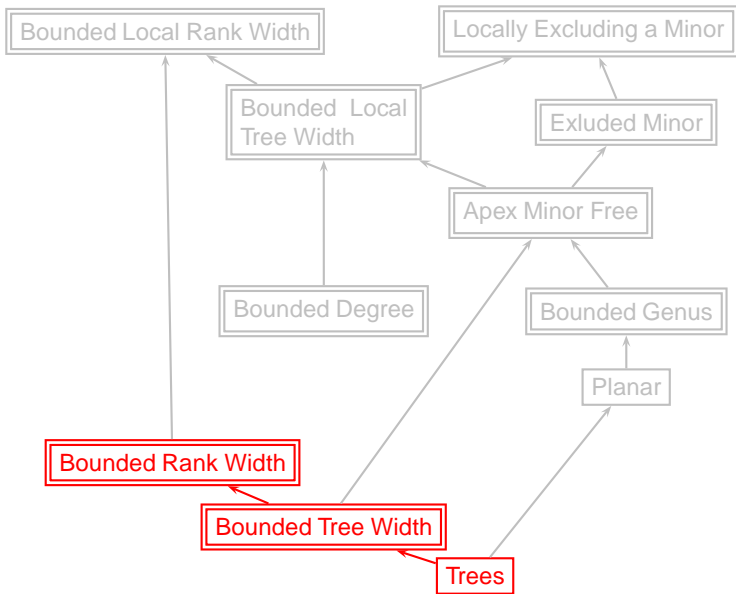
Localisations of Graph Invariants

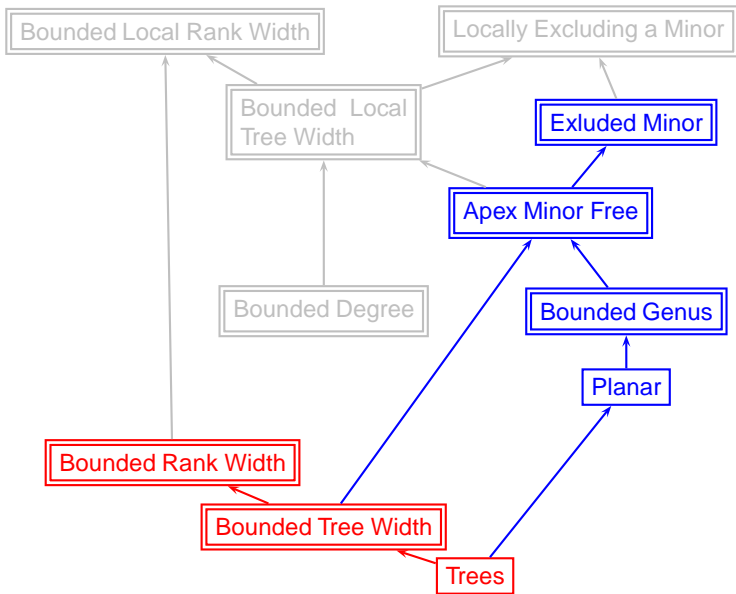
Graph Minor Theory

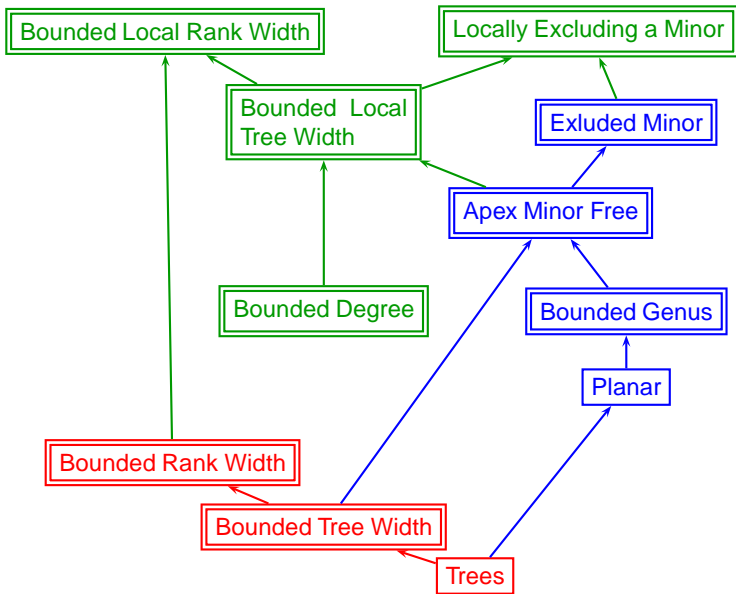
3. Techniques from Logic

4. An Algorithmic Meta Theorem









2. Classes of Graphs

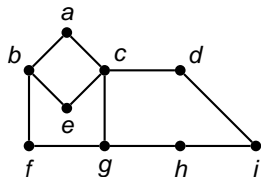
Tree Decompositions and Tree Width

Localisations of Graph Invariants

Graph Minor Theory

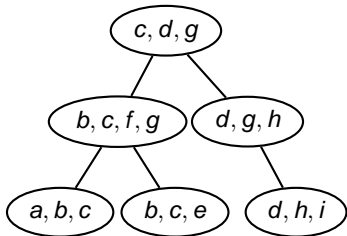
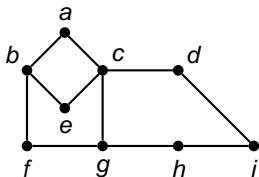
Tree Decompositions

Tree decomposition of a graph:



Tree Decompositions

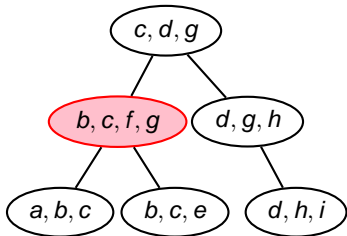
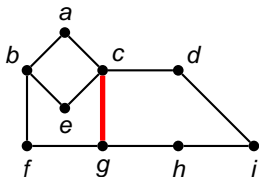
Tree decomposition of a graph:
Family of vertex sets (called **bags**)
arranged in tree, such that:



Tree Decompositions

Tree decomposition of a graph:
Family of vertex sets (called **bags**)
arranged in tree, such that:

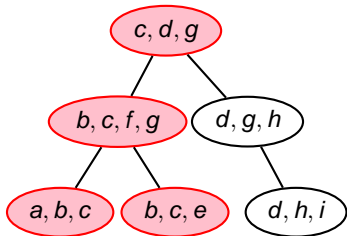
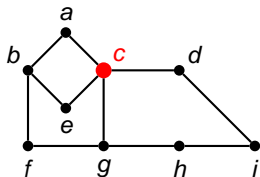
- ▶ every edge is contained in some bag,



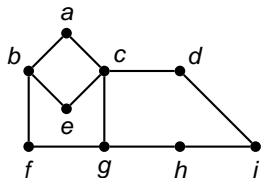
Tree Decompositions

Tree decomposition of a graph:
Family of vertex sets (called **bags**)
arranged in tree, such that:

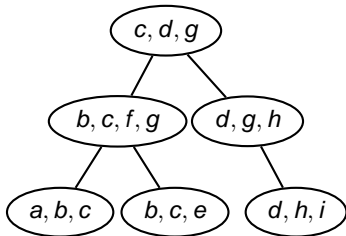
- ▶ every edge is contained in some bag,
- ▶ for every vertex v , the bags that contain v form a (nonempty, connected) subtree



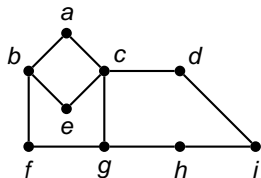
Tree Width



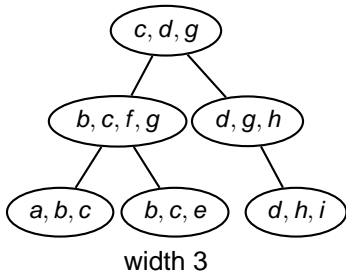
Width of a tree decomposition:
maximum bag size minus 1



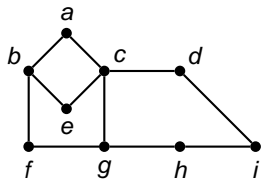
Tree Width



Width of a tree decomposition:
maximum bag size minus 1

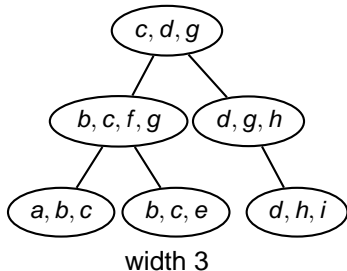


Tree Width

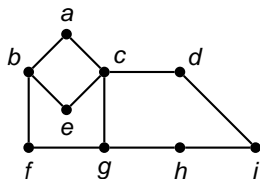


Width of a tree decomposition:
maximum bag size minus 1

Tree width of graph:
minimum width of tree
decomposition

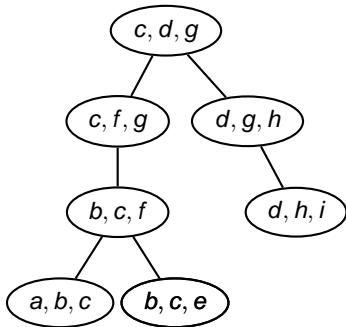


Tree Width



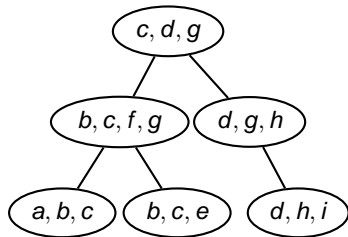
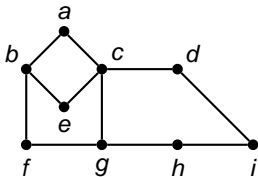
Width of a tree decomposition:
maximum bag size minus 1

Tree width of graph:
minimum width of tree
decomposition

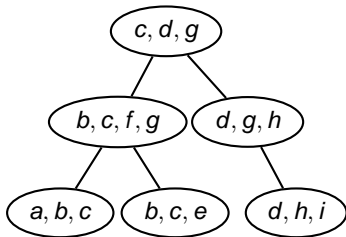
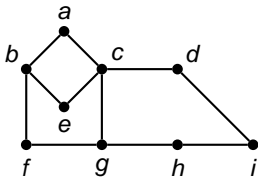


width 2 = tree width

Torsi and decompositions over a class of graphs



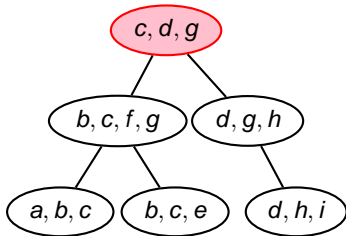
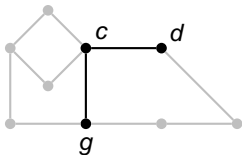
Torsi and decompositions over a class of graphs



Torso at a node of tree:

induced subgraph of bag with additional edges between nodes in intersection with neighbouring bags

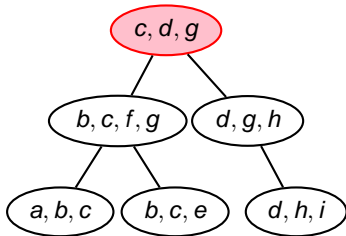
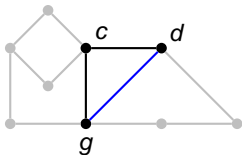
Torsi and decompositions over a class of graphs



Torso at a node of tree:

induced subgraph of bag with additional edges between nodes in intersection with neighbouring bags

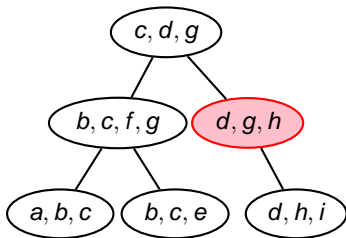
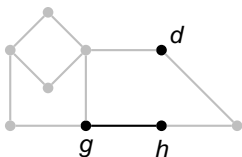
Torsi and decompositions over a class of graphs



Torso at a node of tree:

induced subgraph of bag with additional edges between nodes in intersection with neighbouring bags

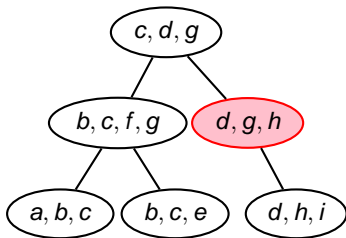
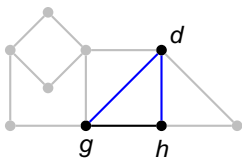
Torsi and decompositions over a class of graphs



Torso at a node of tree:

induced subgraph of bag with additional edges between nodes in intersection with neighbouring bags

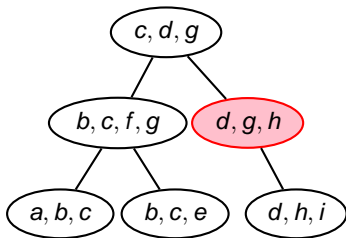
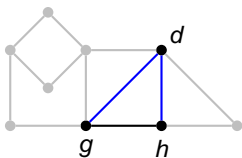
Torsi and decompositions over a class of graphs



Torso at a node of tree:

induced subgraph of bag with additional edges between nodes in intersection with neighbouring bags

Torsi and decompositions over a class of graphs



Torso at a node of tree:

induced subgraph of bag with additional edges between nodes in intersection with neighbouring bags

Tree decomposition is **over** class \mathcal{A} if all torsis belong to \mathcal{A}

2. Classes of Graphs

Tree Decompositions and Tree Width

Localisations of Graph Invariants

Graph Minor Theory

Distances and Neighbourhoods

$G = (V, E)$ graph, $v, w \in V, r \geq 0$

▶ **Distance** between v and w :

$\text{dist}(v, w) :=$ length (# edges) of shortest path from v to w

Distances and Neighbourhoods

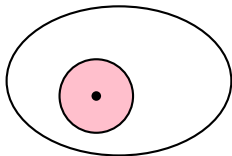
$G = (V, E)$ graph, $v, w \in V, r \geq 0$

- ▶ **Distance** between v and w :

$\text{dist}(v, w) :=$ length (# edges) of shortest path from v to w

- ▶ **r -Neighbourhood** of v :

$$N_r(v) := \{w \mid \text{dist}(v, w) \leq r\}$$



Distances and Neighbourhoods

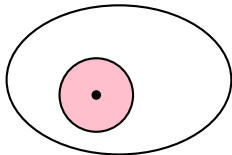
$G = (V, E)$ graph, $v, w \in V, r \geq 0$

- ▶ **Distance** between v and w :

$\text{dist}(v, w) :=$ length (# edges) of shortest path from v to w

- ▶ **r -Neighbourhood** of v :

$$N_r(v) := \{w \mid \text{dist}(v, w) \leq r\}$$



- ▶ $G[N_r(v)]$ induced subgraph of G with vertex set $N_r(v)$

Localisation of Graph Invariants

\mathcal{G} class of all graphs,

$$\mathcal{I} : \mathcal{G} \rightarrow \mathbb{N}$$

invariant under isomorphism

Localisation of Graph Invariants

\mathcal{G} class of all graphs,

$$\mathcal{I} : \mathcal{G} \rightarrow \mathbb{N}$$

invariant under isomorphism

► **Localisation of \mathcal{I} :**

$l_{\mathcal{I}} : \mathcal{G} \times \mathbb{N} \rightarrow \mathbb{N}$ with

$$l_{\mathcal{I}}(\mathbf{G}, r) := \max_{v \in V(\mathbf{G})} \mathcal{I}(\mathbf{G}[N_r(v)]).$$

Localisation of Graph Invariants

\mathcal{G} class of all graphs,

$$\mathcal{I} : \mathcal{G} \rightarrow \mathbb{N}$$

invariant under isomorphism

► **Localisation of \mathcal{I} :**

$\ell_{\mathcal{I}} : \mathcal{G} \times \mathbb{N} \rightarrow \mathbb{N}$ with

$$\ell_{\mathcal{I}}(\mathbf{G}, r) := \max_{v \in V(\mathbf{G})} \mathcal{I}(\mathbf{G}[N_r(v)]).$$

► Class \mathcal{C} has **bounded local \mathcal{I}** if there is a computable $f : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$\ell_{\mathcal{I}}(\mathbf{G}, r) \leq f(r)$$

for all \mathbf{G} and r .

Bounded Local Order

order of a graph = number of vertices

order of a graph = number of vertices

Observation

For every class \mathcal{C} of graphs, the following three statements are equivalent:

order of a graph = number of vertices

Observation

For every class \mathcal{C} of graphs, the following three statements are equivalent:

1. \mathcal{C} has bounded degree.

order of a graph = number of vertices

Observation

For every class \mathcal{C} of graphs, the following three statements are equivalent:

- 1. \mathcal{C} has bounded degree.*
- 2. \mathcal{C} has bounded local order.*

order of a graph = number of vertices

Observation

For every class \mathcal{C} of graphs, the following three statements are equivalent:

- 1. \mathcal{C} has bounded degree.*
- 2. \mathcal{C} has bounded local order.*
- 3. \mathcal{C} has bounded local \mathcal{I} for every graph invariant $\mathcal{I} : \mathcal{G} \rightarrow \mathbb{N}$.*

Local Tree Width of Planar Graphs

Theorem (Robertson, Seymour)

For every planar graph G and every $r \geq 1$:

$$\ell_{\text{tree width}}(G, r) \leq 3r.$$

Local Tree Width of Planar Graphs

Theorem (Robertson, Seymour)

For every planar graph G and every $r \geq 1$:

$$\ell_{\text{tree width}}(G, r) \leq 3r.$$

Hence the class of planar graphs has bounded local tree width.

Example

All classes of graphs with one of the following properties have bounded local tree width:

- ▶ bounded degree,
- ▶ bounded tree width,
- ▶ bounded genus.

2. Classes of Graphs

Tree Decompositions and Tree Width

Localisations of Graph Invariants

Graph Minor Theory

G, H graphs,

- ▶ G is a **minor** of H ($G \preceq H$) if G obtained from H by deleting edges and vertices and **contracting** edges



G, H graphs,

- ▶ G is a **minor** of H ($G \preceq H$) if G obtained from H by deleting edges and vertices and **contracting** edges



- ▶ H **excluded** from G if $H \not\preceq G$.

$$\mathcal{X}(H) := \{G \mid H \text{ excluded from } G\}$$

G, H graphs,

- ▶ G is a **minor** of H ($G \preceq H$) if G obtained from H by deleting edges and vertices and **contracting** edges



- ▶ H **excluded** from G if $H \not\preceq G$.

$$\mathcal{X}(H) := \{G \mid H \text{ excluded from } G\}$$

For class \mathcal{H} :

$$\mathcal{X}(\mathcal{H}) := \{G \mid \forall H \in \mathcal{H} : H \text{ excluded from } G\}.$$

Graph Minor Theorem

Theorem (Robertson and Seymour 2004)

For every class \mathcal{C} that is closed under taking minors there is a finite class \mathcal{F} of graphs such that

$$\mathcal{C} = \mathcal{X}(\mathcal{F})$$

MINOR-TEST

Instance: Graph G, H .

Problem: Decide if $H \preceq G$.

MINOR-TEST

Instance: Graph G, H .

Problem: Decide if $H \preceq G$.

Theorem (Robertson and Seymour 1995)

MINOR-TEST *is solvable in time*

$$f(|H|) \cdot |G|^3,$$

for some computable function f .

MINOR-TEST

Instance: Graph G, H .

Problem: Decide if $H \preceq G$.

Theorem (Robertson and Seymour 1995)

MINOR-TEST *is solvable in time*

$$f(|H|) \cdot |G|^3,$$

for some computable function f .

Corollary

Every class \mathcal{C} closed under taking minors has a cubic time membership test.

Theorem (Robertson and Seymour 1999)

\mathcal{C} class of graphs with excluded minor.

Then all graphs in \mathcal{C} have a tree decomposition over graphs that are “almost embeddable” into some fixed surface.

Structure Simplified

For all $\lambda, \mu \geq 0$:

$$\mathcal{L}(\lambda) := \{G \mid \forall H \preceq G, r \geq 0 : \ell_{\text{tree width}}(H, r) \leq \lambda \cdot r\},$$

For all $\lambda, \mu \geq 0$:

$$\mathcal{L}(\lambda) := \{G \mid \forall H \preceq G, r \geq 0 : \ell_{\text{tree width}}(H, r) \leq \lambda \cdot r\},$$

$$\mathcal{L}(\lambda, \mu) := \{G \mid \exists X \subseteq V(G) : |X| \leq \mu \text{ and } G \setminus X \in \mathcal{L}(\lambda)\}.$$

For all $\lambda, \mu \geq 0$:

$$\mathcal{L}(\lambda) := \{G \mid \forall H \preceq G, r \geq 0 : \ell_{\text{tree width}}(H, r) \leq \lambda \cdot r\},$$

$$\mathcal{L}(\lambda, \mu) := \{G \mid \exists X \subseteq V(G) : |X| \leq \mu \text{ and } G \setminus X \in \mathcal{L}(\lambda)\}.$$

Theorem (G. 2003)

For every class \mathcal{C} with an excluded minor there are $\lambda, \mu \geq 0$ such that all graphs in \mathcal{C} have a tree decomposition over $\mathcal{L}(\lambda, \mu)$.

1. Directions and Results
2. Classes of Graphs
3. Techniques from Logic
 - Decomposition Lemmas
 - Syntactical Interpretations
 - Locality
4. An Algorithmic Meta Theorem

Three Basic Techniques

- ▶ **Composition Lemmas:**
Lift definability from parts of a structure to the whole structure

Three Basic Techniques

- ▶ **Composition Lemmas:**
Lift definability from parts of a structure to the whole structure
- ▶ **Syntactical Interpretations:**
Transfer definability to structure defined within another structure

Three Basic Techniques

- ▶ **Composition Lemmas:**
Lift definability from parts of a structure to the whole structure
- ▶ **Syntactical Interpretations:**
Transfer definability to structure defined within another structure
- ▶ **Locality:**
Lift definability from local neighbourhoods to the whole structure

3. Techniques from Logic

Decomposition Lemmas

Syntactical Interpretations

Locality

Let L be one of FO, MSO, GSO

Let L be one of FO, MSO, GSO

- ▶ **Quantifier rank** $qr(\varphi)$ of L-formula φ :
maximum nesting depth of quantifiers in φ

Let L be one of FO, MSO, GSO

- ▶ **Quantifier rank** $qr(\varphi)$ of L-formula φ :
maximum nesting depth of quantifiers in φ

A structure, \vec{a} tuple of elements, $q \geq 0$:

- ▶ **q -type** of \vec{a} :

$$tp_q(\vec{a}) := \{\varphi(\vec{x}) \mid qr(\varphi) \leq q, A \models \varphi(\vec{a})\}.$$

Remark

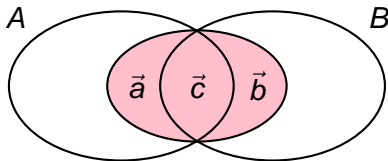
Types are finite up to logical equivalence, and they have canonical finite representations.

Decomposition Lemma

Lemma (Feferman and Vaught)

A, B relational structures and $\vec{a} \in A^k, \vec{b} \in B^\ell,$

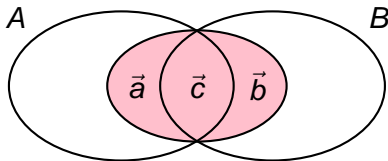
$\vec{c} = (c_1, \dots, c_m)$ such that $A \cap B = \{c_1, \dots, c_m\}$.



Decomposition Lemma

Lemma (Feferman and Vaught)

A, B relational structures and $\vec{a} \in A^k, \vec{b} \in B^\ell,$
 $\vec{c} = (c_1, \dots, c_m)$ such that $A \cap B = \{c_1, \dots, c_m\}$.

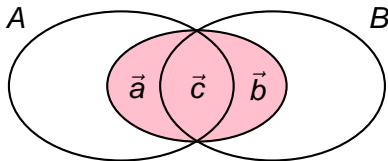


Then $\text{tp}_q(A \cup B, \vec{a}\vec{b}\vec{c})$ is determined by $\text{tp}_q(A, \vec{a}\vec{c})$ and $\text{tp}_q(B, \vec{b}\vec{c})$.

Decomposition Lemma

Lemma (Feferman and Vaught)

A, B relational structures and $\vec{a} \in A^k, \vec{b} \in B^\ell,$
 $\vec{c} = (c_1, \dots, c_m)$ such that $A \cap B = \{c_1, \dots, c_m\}$.



Then $\text{tp}_q(A \cup B, \vec{a}\vec{b}\vec{c})$ is determined by $\text{tp}_q(A, \vec{a}\vec{c})$ and $\text{tp}_q(B, \vec{b}\vec{c})$.

Furthermore, the mapping

$$(\text{tp}_q(A, \vec{a}\vec{c}), \text{tp}_q(B, \vec{b}\vec{c})) \mapsto \text{tp}_q(A \cup B, \vec{a}\vec{b}\vec{c})$$

is computable.

Corollary (Büchi, Elgot, Trakhtenbrot, Thatcher, Wright)

For every L-sentence φ there is a word/tree automaton A_φ accepting precisely the words/trees satisfying φ .

Corollary (Büchi, Elgot, Trakhtenbrot, Thatcher, Wright)

For every L-sentence φ there is a word/tree automaton A_φ accepting precisely the words/trees satisfying φ .

Furthermore, the mapping

$$\varphi \mapsto A_\varphi$$

is computable.

3. Techniques from Logic

Decomposition Lemmas

Syntactical Interpretations

Locality

Syntactical Interpretations

A **syntactical interpretation** Γ is a tuple of formulas that, applied to a structure A , yields a structure $\Gamma(A)$.

Syntactical Interpretations

A **syntactical interpretation** Γ is a tuple of formulas that, applied to a structure A , yields a structure $\Gamma(A)$.

Lemma on Syntactical Interpretations

For every sentence φ there is a sentence $\delta(\varphi)$ such that

$$\Gamma(A) \models \varphi \iff A \models \delta(\varphi).$$

Syntactical Interpretations

A **syntactical interpretation** Γ is a tuple of formulas that, applied to a structure A , yields a structure $\Gamma(A)$.

Lemma on Syntactical Interpretations

For every sentence φ there is a sentence $\delta(\varphi)$ such that

$$\Gamma(A) \models \varphi \iff A \models \delta(\varphi).$$

Furthermore, the mapping

$$\varphi \mapsto \delta(\varphi)$$

is computable.

3. Techniques from Logic

Decomposition Lemmas

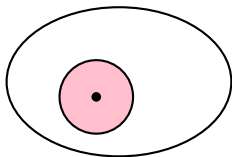
Syntactical Interpretations

Locality

- ▶ **Distances** and **neighbourhoods** in a structure defined with respect to underlying graph (Gaifman graph, primal graph)

- ▶ **Distances** and **neighbourhoods** in a structure defined with respect to underlying graph (Gaifman graph, primal graph)
- ▶ Formula $\varphi(x)$ **r -local** if for all structures A , $a \in A$

$$A \models \varphi(v) \iff A[N_r(v)] \models \varphi(v).$$



Gaifman's Theorem

Basic local sentence:

$$\exists x_1 \dots \exists x_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(x_i, x_j) > 2r \wedge \bigwedge_{i=1}^k \varphi(x_i) \right),$$

where $\varphi(x)$ is r -local.

Gaifman's Theorem

Basic local sentence:

$$\exists x_1 \dots \exists x_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(x_i, x_j) > 2r \wedge \bigwedge_{i=1}^k \varphi(x_i) \right),$$

where $\varphi(x)$ is r -local.

Theorem (Gaifman 1981)

Every first-order sentence φ is equivalent to a Boolean combination λ_φ of basic local sentences.

Gaifman's Theorem

Basic local sentence:

$$\exists x_1 \dots \exists x_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(x_i, x_j) > 2r \wedge \bigwedge_{i=1}^k \varphi(x_i) \right),$$

where $\varphi(x)$ is r -local.

Theorem (Gaifman 1981)

Every first-order sentence φ is equivalent to a Boolean combination λ_φ of basic local sentences.

Furthermore, the mapping

$$\varphi \mapsto \lambda_\varphi$$

is computable.

1. Directions and Results

2. Classes of Graphs

3. Techniques from Logic

4. **An Algorithmic Meta Theorem**

First Step: Structures of Bounded Tree Width

Second Step: Exploit Locality

First-Order Model Checking

\mathcal{C} class of graphs, L logic

L-MODEL-CHECKING ON \mathcal{C}

Instance: Graph $G \in \mathcal{C}$, sentence $\varphi \in L$.

Problem: Decide if $G \models \varphi$.

First-Order Model Checking

\mathcal{C} class of graphs, L logic

L-MODEL-CHECKING ON \mathcal{C}

Instance: Graph $G \in \mathcal{C}$, sentence $\varphi \in L$.

Problem: Decide if $G \models \varphi$.

Theorem (Frick, G. 2001)

Let \mathcal{C} be a class of graphs of bounded local tree width.

FO-MODEL-CHECKING ON \mathcal{C} is solvable in time

$$f(|\varphi|) \cdot |G|^2,$$

for some computable function f .

First-Order Model Checking

\mathcal{C} class of graphs, L logic

L-MODEL-CHECKING ON \mathcal{C}

Instance: Graph $G \in \mathcal{C}$, sentence $\varphi \in L$.

Problem: Decide if $G \models \varphi$.

Theorem (Frick, G. 2001)

Let \mathcal{C} be a class of graphs of bounded local tree width.

FO-MODEL-CHECKING ON \mathcal{C} is solvable in time

$$f(|\varphi|) \cdot |G|^2,$$

for some computable function f .

Corollary

FO-definable properties of planar graphs are decidable in quadratic time.

First-Order Model Checking

Theorem (Frick, G. 2001)

Let \mathcal{C} be a class of graphs of bounded local tree width.

FO-MODEL-CHECKING ON \mathcal{C} is solvable in time

$$f(|\varphi|) \cdot |G|^2,$$

for some computable function f .

4. An Algorithmic Meta Theorem

First Step: Structures of Bounded Tree Width

Second Step: Exploit Locality

Theorem (Courcelle 1990)

GSO-MODEL-CHECKING (*on the class of all graphs*) can be solved in time

$$g(|\varphi|, \text{tree width of } G) \cdot |G|,$$

for some computable function g .

Computing the Decompositions

TREE-DECOMPOSITION

Instance: Graph G .

Problem: Compute tree decomposition of G of minimum width.

Computing the Decompositions

TREE-DECOMPOSITION

Instance: Graph G .

Problem: Compute tree decomposition of G of minimum width.

Theorem (Bodlaender 1996)

TREE-DECOMPOSITION *is solvable in time*

$$2^{O((\text{tree width of } G)^3)} \cdot |G|.$$

MSO Model Checking

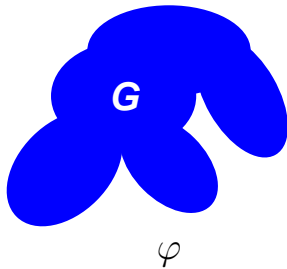
Algorithm *MSO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{MSO}$

MSO Model Checking

Algorithm *MSO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{MSO}$

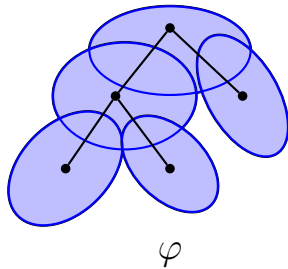


MSO Model Checking

Algorithm *MSO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{MSO}$

1. Compute minimum width tree decomposition of G .

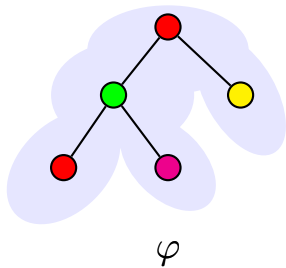


MSO Model Checking

Algorithm *MSO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{MSO}$

1. Compute minimum width tree decomposition of G .
2. Interpret G in tree underlying the decomposition.

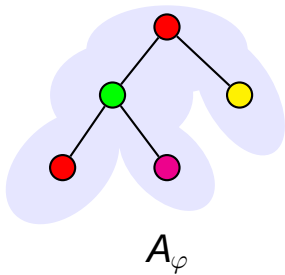


MSO Model Checking

Algorithm *MSO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{MSO}$

1. Compute minimum width tree decomposition of G .
2. Interpret G in tree underlying the decomposition.
3. Translate φ into tree-automaton A_φ and run it.



4. An Algorithmic Meta Theorem

First Step: Structures of Bounded Tree Width

Second Step: Exploit Locality

FO Model Checking

Algorithm *FO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{FO}$

Algorithm *FO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{FO}$

1. Translate φ into BK λ_φ of basic local sentences.

Algorithm *FO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{FO}$

1. Translate φ into BK λ_φ of basic local sentences.
2. For every

$$\psi = \exists \mathbf{x}_1 \dots \exists \mathbf{x}_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(\mathbf{x}_i, \mathbf{x}_j) > 2r \wedge \bigwedge_{i=1}^k \chi(\mathbf{x}_i) \right)$$

in λ_φ , do the following:

Algorithm *FO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{FO}$

1. Translate φ into BK λ_φ of basic local sentences.
2. For every

$$\psi = \exists \mathbf{x}_1 \dots \exists \mathbf{x}_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(\mathbf{x}_i, \mathbf{x}_j) > 2r \wedge \bigwedge_{i=1}^k \chi(\mathbf{x}_i) \right)$$

in λ_φ , do the following:

3. Compute set of all $v \in V(G)$ such that $G[N_r(v)] \models \chi(v)$ and colour them **red**.

Algorithm *FO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{FO}$

1. Translate φ into BK λ_φ of basic local sentences.
2. For every

$$\psi = \exists \mathbf{x}_1 \dots \exists \mathbf{x}_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(\mathbf{x}_i, \mathbf{x}_j) > 2r \wedge \bigwedge_{i=1}^k \chi(\mathbf{x}_i) \right)$$

in λ_φ , do the following:

3. Compute set of all $v \in V(G)$ such that $G[N_r(v)] \models \chi(v)$ and colour them **red**.
4. Check if there are k red vertices of pairwise distance $> 2r$.

Algorithm *FO-Model-Checking*

Input: Graph G , sentence $\varphi \in \text{FO}$

1. Translate φ into BK λ_φ of basic local sentences.
2. For every

$$\psi = \exists \mathbf{x}_1 \dots \exists \mathbf{x}_k \left(\bigwedge_{1 \leq i < j \leq k} \text{dist}(\mathbf{x}_i, \mathbf{x}_j) > 2r \wedge \bigwedge_{i=1}^k \chi(\mathbf{x}_i) \right)$$

in λ_φ , do the following:

3. Compute set of all $v \in V(G)$ such that $G[N_r(v)] \models \chi(v)$ and colour them **red**.
4. Check if there are k red vertices of pairwise distance $> 2r$.
5. Combine results to evaluate λ_φ .

RED-VERTICES

Instance: A graph $G = (V, E)$, a set $R \subseteq V$, and $k, r \geq 0$.

Problem: Decide if there are $v_1, \dots, v_k \in R$ such that $\text{dist}(v_i, v_j) > 2r$ for all $i \neq j$.

RED-VERTICES

Instance: A graph $G = (V, E)$, a set $R \subseteq V$, and $k, r \geq 0$.

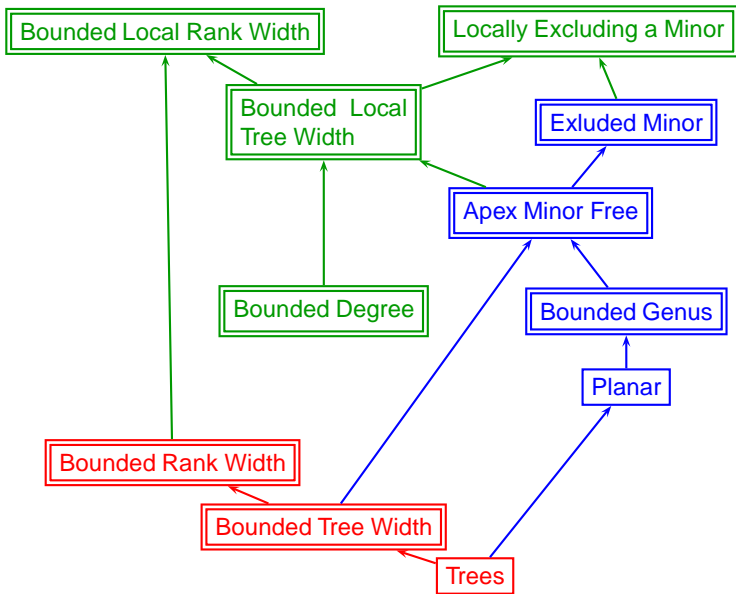
Problem: Decide if there are $v_1, \dots, v_k \in R$ such that $\text{dist}(v_i, v_j) > 2r$ for all $i \neq j$.

Lemma

RED VERTICES *is solvable in time*

$$h(k, \ell_{\text{tree width}}(G, r)) \cdot |G|$$

for some computable function h .



Further Interesting Classes

- ▶ classes of graphs closed under taking induced subgraphs or induced minors (interval graphs, chordal graphs, perfect graphs, claw free graphs, graphs of bounded rank width, etc)

Further Interesting Classes

- ▶ classes of graphs closed under taking induced subgraphs or induced minors (interval graphs, chordal graphs, perfect graphs, claw free graphs, graphs of bounded rank width, etc)
- ▶ classes of graphs of bounded expansion

Further Interesting Classes

- ▶ classes of graphs closed under taking induced subgraphs or induced minors (interval graphs, chordal graphs, perfect graphs, claw free graphs, graphs of bounded rank width, etc)
- ▶ classes of graphs of bounded expansion
- ▶ algebraically defined classes of structures (groups, fields, lattices, etc)

A survey on algorithmic meta theorems entitled

Logic, Graphs, and Algorithms

can be found on my web page:

<http://www.informatik.hu-berlin.de/~grohe/pub/meta-survey.pdf>