Definable Structure Theory

Martin Grohe
Humboldt-Universität zu Berlin

Outline

1. Graph Structure Theory
2. Treelike Decompositions
3. Two Applications
4. Open Problems

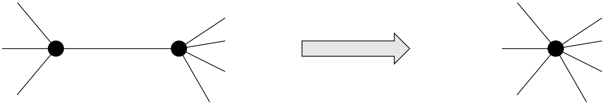
Graph Structure Theory

Graph Minors

Graph G is a minor of graph H
if G is obtained from H by deleting edges and vertices and contracting edges.

Graph Minors

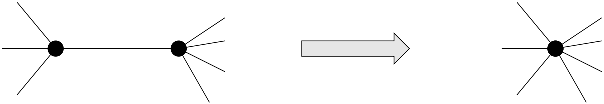
Graph G is a minor of graph H
if G is obtained from H by deleting edges and vertices and contracting edges.



Contracting an edge

Graph Minors

Graph G is a minor of graph H
if G is obtained from H by deleting edges and vertices and contracting edges.



Contracting an edge

H is excluded minor for class $\mathcal{C}$ of graphs
if H is not a minor of any graph in $\mathcal{C}$

Examples of Classes with Excluded Minors

- The class of planar graphs and all classes of graphs embeddable in a fixed surface

Examples of Classes with Excluded Minors

- The class of planar graphs and all classes of graphs embeddable in a fixed surface
- The classes of graphs knotlessly or linklessly embeddable in 3-space

Examples of Classes with Excluded Minors

- The class of planar graphs and all classes of graphs embeddable in a fixed surface
- The classes of graphs knotlessly or linklessly embeddable in 3-space
- The class of all graphs with a vertex cover of size at most k.

Examples of Classes with Excluded Minors

- The class of planar graphs and all classes of graphs embeddable in a fixed surface
- The classes of graphs knotlessly or linklessly embeddable in 3-space
- The class of all graphs with a vertex cover of size at most k.
- The class of all graphs with a feedback vertex set size at most k.

Examples of Classes with Excluded Minors

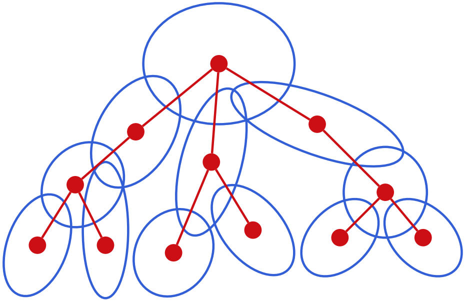
- The class of planar graphs and all classes of graphs embeddable in a fixed surface
- The classes of graphs knotlessly or linklessly embeddable in 3-space
- The class of all graphs with a vertex cover of size at most k.
- The class of all graphs with a feedback vertex set size at most k.
- The class of apex graphs (planar plus one vertex) and all classes obtained from a class with excluded minors by adding a bounded number of vertices.

Examples of Classes with Excluded Minors

- The class of planar graphs and all classes of graphs embeddable in a fixed surface
- The classes of graphs knotlessly or linklessly embeddable in 3-space
- The class of all graphs with a vertex cover of size at most k.
- The class of all graphs with a feedback vertex set size at most k.
- The class of apex graphs (planar plus one vertex) and all classes obtained from a class with excluded minors by adding a bounded number of vertices.
- All classes of graphs of bounded tree width.

A tree decomposition decomposes a graph into pieces arranged in a treelike fashion.

A tree decomposition decomposes a graph into pieces arranged in a treelike fashion.



The Structure of Graphs with Excluded Minors

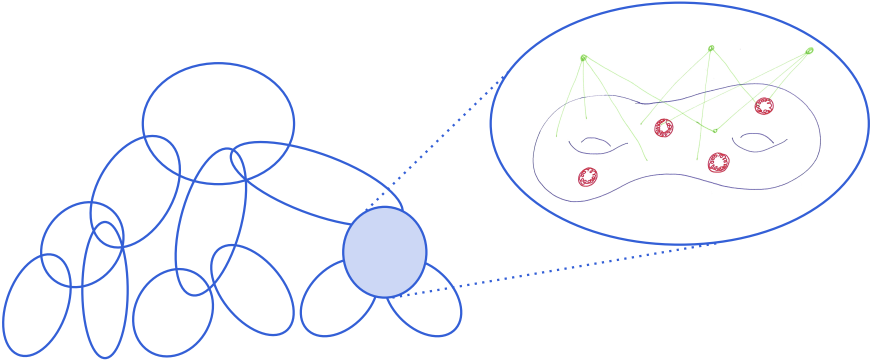
Theorem (Robertson and Seymour 1999)

Let $\mathcal{C}$ be a class of graphs with an excluded minor. Then all graphs in $\mathcal{C}$ have a tree decomposition into pieces that are almost embeddable in some surface.

The Structure of Graphs with Excluded Minors

Theorem (Robertson and Seymour 1999)

Let $\mathcal{C}$ be a class of graphs with an excluded minor. Then all graphs in $\mathcal{C}$ have a tree decomposition into pieces that are almost embeddable in some surface.



The Structure of Graphs with Excluded Minors

Theorem (Robertson and Seymour 1999)

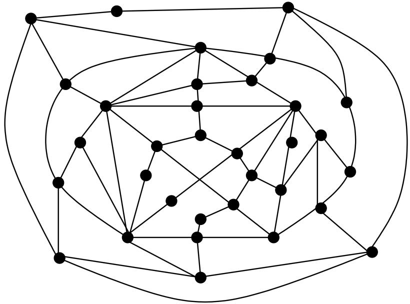
Let $\mathcal{C}$ be a class of graphs with an excluded minor. Then all graphs in $\mathcal{C}$ have a tree decomposition into pieces that are almost embeddable in some surface.

Moreover, there is an algorithm that computes such a decomposition in time $f(k)n^3$, where k is the size of the excluded minor and n the size of the decomposed graph (Kawarabayashi, Wollan 2011).

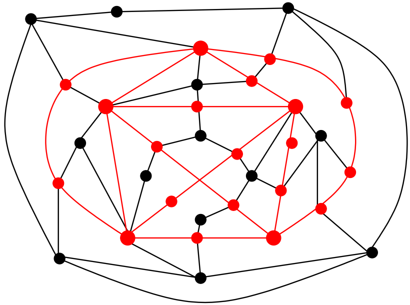
Topological Subgraphs

G is a topological subgraph of H
if a subdivision of G is a subgraph of H.

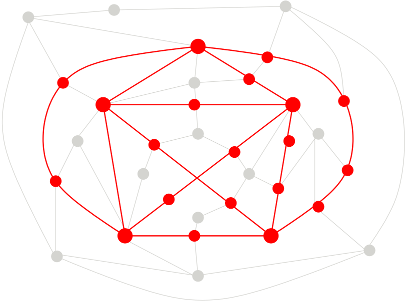
G is a topological subgraph of H
if a subdivision of G is a subgraph of H.



G is a topological subgraph of H
if a subdivision of G is a subgraph of H.

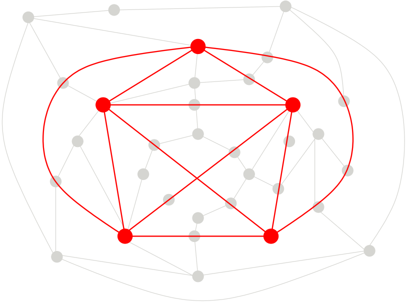


G is a topological subgraph of H
if a subdivision of G is a subgraph of H.

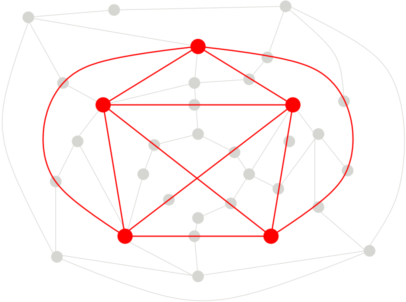


Topological Subgraphs

G is a topological subgraph of H
if a subdivision of G is a subgraph of H.



G is a topological subgraph of H
if a subdivision of G is a subgraph of H.



Observation

If G is a topological subgraph of H then it is a minor of H. In general, the converse does not hold.

Excluding Topological Subgraphs

H is excluded topological subgraph for class $\mathcal{C}$ of graphs
if H is not a topological subgraph of any graph in $\mathcal{C}$

Excluding Topological Subgraphs

H is excluded topological subgraph for class $\mathcal{C}$ of graphs
if H is not a topological subgraph of any graph in $\mathcal{C}$

Examples of Classes with Excluded Topological Subgraphs

- All classes with excluded minors.

Excluding Topological Subgraphs

H is excluded topological subgraph for class $\mathcal{C}$ of graphs
if H is not a topological subgraph of any graph in $\mathcal{C}$

Examples of Classes with Excluded Topological Subgraphs

- All classes with excluded minors.
- All classes of bounded degree.

The Structure of Graphs with Excluded Topological Subgraphs

Theorem (G. and Marx 2012)

Let $\mathcal{C}$ be a class of graphs with an excluded topological subgraph. Then all graphs $G \in \mathcal{C}$ have a tree decomposition into pieces that are

- *either almost embeddable in some surface,*
- *or of bounded degree after removing a bounded number of vertices.*

The Structure of Graphs with Excluded Topological Subgraphs

Theorem (G. and Marx 2012)

Let $\mathcal{C}$ be a class of graphs with an excluded topological subgraph. Then all graphs $G \in \mathcal{C}$ have a tree decomposition into pieces that are

- *either almost embeddable in some surface,*
- *or of bounded degree after removing a bounded number of vertices.*

Moreover, there is an algorithm that computes such a decomposition in time $f(k)n^{O(1)}$.

Structure Theory for Classes Closed Under Induced Subgraphs

- Chordal graphs and subclasses such as interval graphs
- Perfect graphs and claw-free graphs (Chudnowsky, Seymour)
- Graphs of bounded rank width (or clique width)

Treelike Decompositions

Question

Can we define the decompositions of the structure theorems in a reasonable logic ?

Answer

No!

Question

Why not?

Question
Why not?

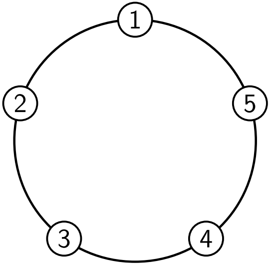
Answer
Because tree decompositions are not invariant under automorphisms.

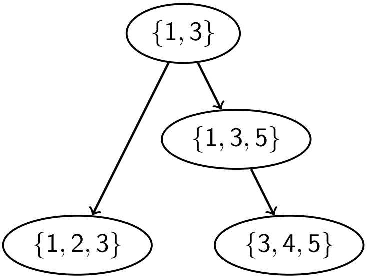
Example

A tree decomposition of the cycle C_5.

Example

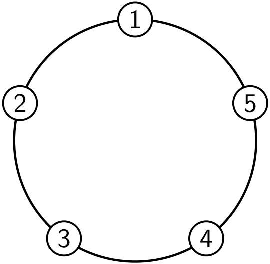
A tree decomposition of the cycle C_5.

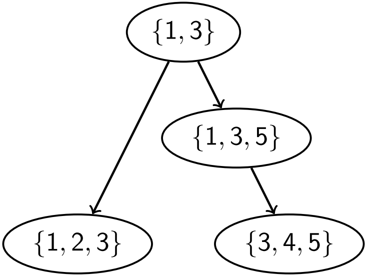




Example

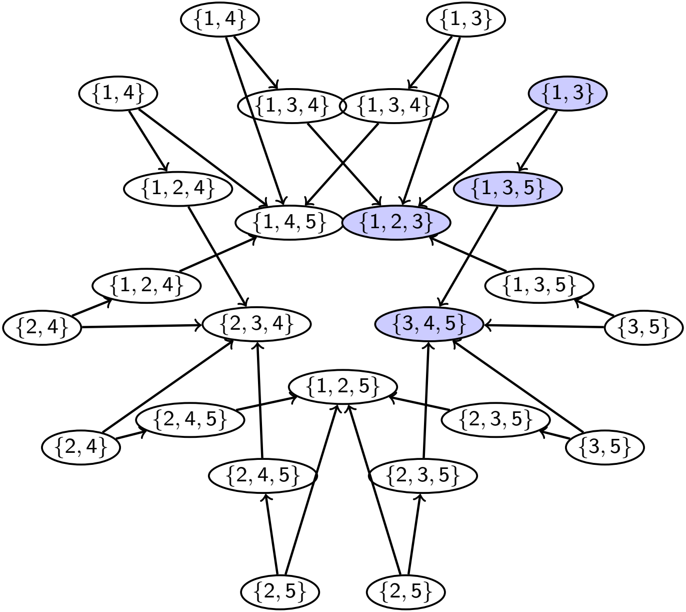
A tree decomposition of the cycle C_5.





Not automorphism-invariant:
Automorphism $i \mapsto (i+1) \mod 5$ does not extend to decomposition

Closure Under Automorphisms



A tree decomposition of a graph G is a pair (T, β), where T is a tree and $\beta : V(T) \to 2^{V(G)}$ such that:

(T1) For all $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta(t)\}$ is connected in T.

(T2) For all $vw \in E(G)$ there is a $t \in V(T)$ such that $v, w \in \beta(t)$.

A tree decomposition of a graph G is a pair (T, β), where T is a tree and $\beta : V(T) \to 2^{V(G)}$ such that:

(T1) For all $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta(t)\}$ is connected in T.

(T2) For all $vw \in E(G)$ there is a $t \in V(T)$ such that $v, w \in \beta(t)$.

Without loss of generality T directed.

Tree Decompositions

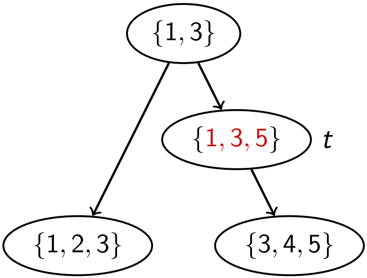
A tree decomposition of a graph G is a pair (T, β), where T is a tree and $\beta : V(T) \to 2^{V(G)}$ such that:

(T1) For all $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta(t)\}$ is connected in T.

(T2) For all $vw \in E(G)$ there is a $t \in V(T)$ such that $v, w \in \beta(t)$.

Without loss of generality T directed.

Bag at node t: $\beta(t)$



A tree decomposition of a graph G is a pair (T, β), where T is a tree and $\beta : V(T) \to 2^{V(G)}$ such that:

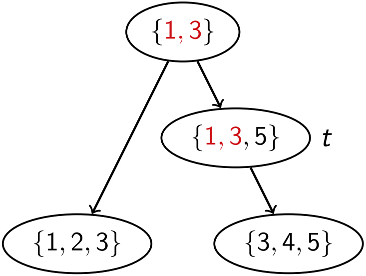
(T1) For all $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta(t)\}$ is connected in T.

(T2) For all $vw \in E(G)$ there is a $t \in V(T)$ such that $v, w \in \beta(t)$.

Without loss of generality T directed.

Bag at node t: $\beta(t)$

Separator at t: $\sigma(t) := \beta(t) \cap \beta(s)$
for parent s of t or $\emptyset$ if t root



A tree decomposition of a graph G is a pair (T, β), where T is a tree and $\beta : V(T) \to 2^{V(G)}$ such that:

(T1) For all $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta(t)\}$ is connected in T.

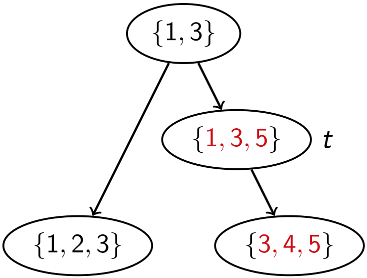
(T2) For all $vw \in E(G)$ there is a $t \in V(T)$ such that $v, w \in \beta(t)$.

Without loss of generality T directed.

Bag at node t: $\beta(t)$

Separator at t: $\sigma(t) := \beta(t) \cap \beta(s)$ for parent s of t or $\emptyset$ if t root

Cone at t: $\gamma(t) := \beta(t) \cup \bigcup_u \beta(u)$ where union ranges over all descendants u of t



Tree Decompositions

A tree decomposition of a graph G is a pair (T, β), where T is a tree and $\beta : V(T) \to 2^{V(G)}$ such that:

(T1) For all $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta(t)\}$ is connected in T.

(T2) For all $vw \in E(G)$ there is a $t \in V(T)$ such that $v, w \in \beta(t)$.

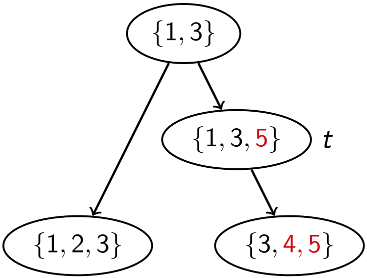
Without loss of generality T directed.

Bag at node t: $\beta(t)$

Separator at t: $\sigma(t) := \beta(t) \cap \beta(s)$ for parent s of t or $\emptyset$ if t root

Cone at t: $\gamma(t) := \beta(t) \cup \bigcup_u \beta(u)$ where union ranges over all descendants u of t

Component at t: $\alpha(t) := \gamma(t) \setminus \sigma(t)$.



Decomposition Based on Separators and Components

A decomposition of a graph G is a triple (D, σ, α), where D is a directed graph and $\sigma, \alpha : V(D) \to 2^{V(G)}$.

Decomposition Based on Separators and Components

A decomposition of a graph G is a triple (D, σ, α), where D is a directed graph and $\sigma, \alpha : V(D) \to 2^{V(G)}$. For all $t \in V(D)$,

$$\gamma(t) := \sigma(t) \cup \alpha(t) \qquad \beta(t) := \gamma(t) \setminus \bigcup_{u \in N_+^D(t)} \alpha(u).$$

Decomposition Based on Separators and Components

A decomposition of a graph G is a triple (D, σ, α), where D is a directed graph and $\sigma, \alpha : V(D) \to 2^{V(G)}$. For all $t \in V(D)$,

$$\gamma(t) := \sigma(t) \cup \alpha(t) \qquad \beta(t) := \gamma(t) \setminus \bigcup_{u \in N_+^D(t)} \alpha(u).$$

Proposition

(D, β) is a tree decomposition iff

(T'1) *D is a tree;*

(T'2) *$\sigma(t) \cap \alpha(t) = \emptyset$ and $N^G(\alpha(t)) \subseteq \sigma(t)$ for all $t \in V(D)$;*

(T'3) *$\alpha(t) \supseteq \alpha(u)$ and $\gamma(t) \supseteq \gamma(u)$ for all $tu \in E(D)$;*

(T'4) *$\gamma(u_1) \cap \gamma(u_2) = \sigma(u_1) \cap \sigma(u_2)$ for all $tu_1, tu_2 \in E(D)$ with $u_1 \neq u_2$;*

(T'5) *$\alpha(r) = V(G)$ for the root r of D.*

Definition

A decomposition (D, σ, α) of a graph G is treelike if

Definition

A decomposition (D, σ, α) of a graph G is treelike if

(TL1) D is acyclic;

Definition

A decomposition (D, σ, α) of a graph G is treelike if

(TL1) D is acyclic;

(TL2) $\sigma(t) \cap \alpha(t) = \emptyset$ and $N^G(\alpha(t)) \subseteq \sigma(t)$ for all $t \in V(D)$;

(TL3) $\alpha(t) \supseteq \alpha(u)$ and $\gamma(t) \supseteq \gamma(u)$ for all $tu \in E(D)$;

Definition

A decomposition (D, σ, α) of a graph G is treelike if

(TL1) D is acyclic;

(TL2) $\sigma(t) \cap \alpha(t) = \emptyset$ and $N^G(\alpha(t)) \subseteq \sigma(t)$ for all $t \in V(D)$;

(TL3) $\alpha(t) \supseteq \alpha(u)$ and $\gamma(t) \supseteq \gamma(u)$ for all $tu \in E(D)$;

(TL4) Either $\gamma(u_1) \cap \gamma(u_2) = \sigma(u_1) \cap \sigma(u_2)$
or $\sigma(u_1) = \sigma(u_2)$ and $\alpha(u_1) = \alpha(u_2)$ for all $tu_1, tu_2 \in E(D)$;

Definition

A decomposition (D, σ, α) of a graph G is treelike if

(TL1) D is acyclic;

(TL2) $\sigma(t) \cap \alpha(t) = \emptyset$ and $N^G(\alpha(t)) \subseteq \sigma(t)$ for all $t \in V(D)$;

(TL3) $\alpha(t) \supseteq \alpha(u)$ and $\gamma(t) \supseteq \gamma(u)$ for all $tu \in E(D)$;

(TL4) Either $\gamma(u_1) \cap \gamma(u_2) = \sigma(u_1) \cap \sigma(u_2)$
or $\sigma(u_1) = \sigma(u_2)$ and $\alpha(u_1) = \alpha(u_2)$ for all $tu_1, tu_2 \in E(D)$;

(TL5) For every connected component A of G there is a node
$t \in V(D)$ such that $\alpha(r) = V(A)$ and $\sigma(t) = \emptyset$.

Definition

1. A decomposition mapping Λ (on a class $\mathcal{C}$ of graphs) maps each graph G (in $\mathcal{C}$) to a decomposition $\Lambda(G)$.

Definition

1. A decomposition mapping Λ (on a class $\mathcal{C}$ of graphs) maps each graph G (in $\mathcal{C}$) to a decomposition $\Lambda(G)$.
2. Λ is treelike if $\Lambda(G)$ is treelike for all G (in $\mathcal{C}$).

Definition

1. A decomposition mapping Λ (on a class $\mathcal{C}$ of graphs) maps each graph G (in $\mathcal{C}$) to a decomposition $\Lambda(G)$.
2. Λ is treelike if $\Lambda(G)$ is treelike for all G (in $\mathcal{C}$).
3. Λ is invariant if for isomorphic G, G' (in $\mathcal{C}$) with $\Lambda(G) = (D, \sigma, \alpha)$, $\Lambda(G') = (D', \sigma', \alpha')$

Decomposition Mappings

Definition

1. A decomposition mapping Λ (on a class $\mathcal{C}$ of graphs) maps each graph G (in $\mathcal{C}$) to a decomposition $\Lambda(G)$.
2. Λ is treelike if $\Lambda(G)$ is treelike for all G (in $\mathcal{C}$).
3. Λ is invariant if for isomorphic G, G' (in $\mathcal{C}$) with $\Lambda(G) = (D, \sigma, \alpha)$, $\Lambda(G') = (D', \sigma', \alpha')$ and for all isomorphisms f from G to G'

Definition

1. A decomposition mapping Λ (on a class $\mathcal{C}$ of graphs) maps each graph G (in $\mathcal{C}$) to a decomposition $\Lambda(G)$.
2. Λ is treelike if $\Lambda(G)$ is treelike for all G (in $\mathcal{C}$).
3. Λ is invariant if for isomorphic G, G' (in $\mathcal{C}$) with $\Lambda(G) = (D, \sigma, \alpha)$, $\Lambda(G') = (D', \sigma', \alpha')$ and for all isomorphisms f from G to G' there is an isomorphism g from D to D' such that

$$f(\sigma(t)) = \sigma(g(t)) \quad \text{and} \quad f(\alpha(t)) = \alpha(g(t))$$

for all $t \in V(D)$.

Definable Decompositions

Definition

Let L be a logic. A decomposition mapping Λ on $\mathcal{C}$ is L-definable if there are L-formulas

$$\lambda_V(\vec{x}), \quad \lambda_E(\vec{x}, \vec{x}'), \quad \lambda_\sigma(\vec{x}, y), \quad \lambda_\alpha(\vec{x}, y)$$

Definition

Let L be a logic. A decomposition mapping Λ on $\mathcal{C}$ is L-definable if there are L-formulas

$$\lambda_V(\vec{x}), \quad \lambda_E(\vec{x}, \vec{x}'), \quad \lambda_\sigma(\vec{x}, y), \quad \lambda_\alpha(\vec{x}, y)$$

such that for all $G \in \mathcal{C}$ with $\Lambda(G) = (D, \sigma, \alpha)$:

- $V(D) = \{\vec{v} \mid G \models \lambda_V[\vec{v}]\}$,

Definable Decompositions

Definition

Let L be a logic. A decomposition mapping Λ on $\mathcal{C}$ is L-definable if there are L-formulas

$$\lambda_V(\vec{x}), \quad \lambda_E(\vec{x}, \vec{x}'), \quad \lambda_\sigma(\vec{x}, y), \quad \lambda_\alpha(\vec{x}, y)$$

such that for all $G \in \mathcal{C}$ with $\Lambda(G) = (D, \sigma, \alpha)$:

- $V(D) = \{\vec{v} \mid G \models \lambda_V[\vec{v}]\}$,
- $E(D) = \{(\vec{v}, \vec{v}') \in V(D)^2 \mid G \models \lambda_E[\vec{v}, \vec{v}']\}$,

Definition

Let L be a logic. A decomposition mapping Λ on $\mathcal{C}$ is L-definable if there are L-formulas

$$\lambda_V(\vec{x}), \quad \lambda_E(\vec{x}, \vec{x}'), \quad \lambda_\sigma(\vec{x}, y), \quad \lambda_\alpha(\vec{x}, y)$$

such that for all $G \in \mathcal{C}$ with $\Lambda(G) = (D, \sigma, \alpha)$:

- $V(D) = \{\vec{v} \mid G \models \lambda_V[\vec{v}]\}$,
- $E(D) = \{(\vec{v}, \vec{v}') \in V(D)^2 \mid G \models \lambda_E[\vec{v}, \vec{v}']\}$,
- $\sigma(\vec{v}) = \{w \mid G \models \lambda_\sigma[\vec{v}, w]\}$ for all $\vec{v} \in V(D)$,

Definition

Let L be a logic. A decomposition mapping Λ on $\mathcal{C}$ is L-definable if there are L-formulas

$$\lambda_V(\vec{x}), \quad \lambda_E(\vec{x}, \vec{x}'), \quad \lambda_\sigma(\vec{x}, y), \quad \lambda_\alpha(\vec{x}, y)$$

such that for all $G \in \mathcal{C}$ with $\Lambda(G) = (D, \sigma, \alpha)$:

- $V(D) = \{\vec{v} \mid G \models \lambda_V[\vec{v}]\}$,
- $E(D) = \{(\vec{v}, \vec{v}') \in V(D)^2 \mid G \models \lambda_E[\vec{v}, \vec{v}']\}$,
- $\sigma(\vec{v}) = \{w \mid G \models \lambda_\sigma[\vec{v}, w]\}$ for all $\vec{v} \in V(D)$,
- $\alpha(\vec{v}) = \{w \mid G \models \lambda_\alpha[\vec{v}, w]\}$ for all $\vec{v} \in V(D)$.

Definition

Let L be a logic. A decomposition mapping Λ on $\mathcal{C}$ is L-definable if there are L-formulas

$$\lambda_V(\vec{x}), \quad \lambda_E(\vec{x}, \vec{x}'), \quad \lambda_\sigma(\vec{x}, y), \quad \lambda_\alpha(\vec{x}, y)$$

such that for all $G \in \mathcal{C}$ with $\Lambda(G) = (D, \sigma, \alpha)$:

- $V(D) = \{\vec{v} \mid G \models \lambda_V[\vec{v}]\}$,
- $E(D) = \{(\vec{v}, \vec{v}') \in V(D)^2 \mid G \models \lambda_E[\vec{v}, \vec{v}']\}$,
- $\sigma(\vec{v}) = \{w \mid G \models \lambda_\sigma[\vec{v}, w]\}$ for all $\vec{v} \in V(D)$,
- $\alpha(\vec{v}) = \{w \mid G \models \lambda_\alpha[\vec{v}, w]\}$ for all $\vec{v} \in V(D)$.

Remark

Definable decomposition mappings are always invariant.

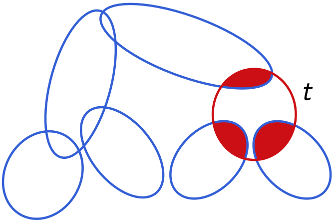
Definition

(D, σ, α) decomposition of G.

The torso at a node $t \in V(D)$ is the graph $\tau(t)$ with

- vertex set $\beta(t)$
- edge set

  $$\{vw \in \beta(t)^2 \mid vw \in E(G) \text{ or } v, w \in \sigma(t) \text{ or } v, w \in \sigma(u) \text{ for some } u \in N_+^D(t)\}.$$



Decompositions into 3-Connected Components

Lemma (G. 2008)

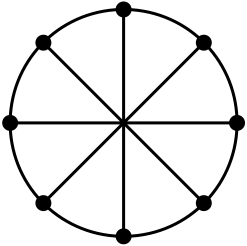
There is an IFP*-definable decomposition mapping* Λ *such that for all* G*, the torsos of* $\Lambda(G)$ *are 3-connected topological subgraphs of* G*.*

Definable Structure Theorem for K_5-Minor Free Graphs

Theorem (G. 2008)

There is an IFP*-definable decomposition mapping* Λ *such that for all* K_5*-minor free graphs* G *the torsos of* $\Lambda(G)$ *are*

- *either* 3*-connected planar graphs*
- *or copies of the graph*



Definable Structure Theorem for Graphs with Excluded Minors

Theorem (G. 2010)

Let $\mathcal{C}$ be a class of graphs with an excluded minor. There is an IFP*-definable decomposition mapping Λ such that for all $G \in \mathcal{C}$ the bags of $\Lambda(G)$ admit an* IFP*-definable linear order.*

Definable Structure Theorem for Graphs with Excluded Minors

Theorem (G. 2010)

Let $\mathcal{C}$ be a class of graphs with an excluded minor. There is an IFP*-definable decomposition mapping Λ such that for all $G \in \mathcal{C}$ the bags of $\Lambda(G)$ admit an* IFP*-definable linear order.*

More precisely, there is an IFP*-formula $\lambda_{\leq}(\vec{x}, y_1, y_2)$ such that for all $G \in \mathcal{C}$ and all nodes $\vec{v}$ of $\Lambda(G)$,*

$$\{(w_1, w_2) \mid G \models \lambda_{\leq}[\vec{v}, w_1, w_2]$$

is a linear order of $\beta(\vec{v})$.

Definable Structure Theorem for Graphs with Excluded Minors

Theorem (G. 2010)

Let $\mathcal{C}$ be a class of graphs with an excluded minor. There is an IFP*-definable decomposition mapping Λ such that for all $G \in \mathcal{C}$ the bags of $\Lambda(G)$ admit an* IFP*-definable linear order.*

More precisely, there is an IFP*-formula $\lambda_{\leq}(\vec{x}, y_1, y_2)$ such that for all $G \in \mathcal{C}$ and all nodes $\vec{v}$ of $\Lambda(G)$,*

$$\{(w_1, w_2) \mid G \models \lambda_{\leq}[\vec{v}, w_1, w_2]\}$$

is a linear order of $\beta(\vec{v})$.

We say that $\mathcal{C}$ admits IFP*-definable ordered treelike decompositions.*

Invariant Structure Theorem for Graphs with Excluded Topological Subgraphs

Theorem (G., Marx 2012)

Let $\mathcal{C}$ be a class of graphs with an excluded topological subgraph. Then there is a polynomial time computable invariant treelike decomposition mapping on $\mathcal{C}$ such that for all $G \in \mathcal{C}$, the torsos of $\Lambda(G)$ are

- *either almost embeddable in some surface,*
- *or of bounded degree after removing a bounded number of vertices.*

Two Applications

Capturing Polynomial Time on Graphs with Excluded Minors

Theorem (G. 2010)

Let $\mathcal{C}$ *be a class of graphs with excluded minors. Then* IFP+C *captures polynomial time on* $\mathcal{C}$*.*

Capturing Polynomial Time on Graphs with Excluded Minors

Theorem (G. 2010)

Let $\mathcal{C}$ be a class of graphs with excluded minors. Then IFP+C *captures polynomial time on $\mathcal{C}$.*

Corollary

Let $\mathcal{C}$ be a class of graphs with excluded minors. Then there is a $k \in \mathbb{N}$ such that the k-dimensional Weisfeiler-Lehman algorithm decides isomorphism of graphs in $\mathcal{C}$ (in polynomial time).

Isomorphism Test for Graphs with Excluded Topological Subgraphs

Theorem (G., Marx 2012)

Let $\mathcal{C}$ be a class of graphs with topological subgraphs. Then there is a polynomial time isomorphism test for graphs in $\mathcal{C}$.

Open Problems

We know:

- The classes of planar graphs and K_5-minor free graphs are IFP-definable (G. 1998, 2008)
- For all k, the class of graphs of tree width at most k is IFP-definable (G., Mariño 1999)
- All nontrivial classes of graphs closed under taking minors are IFP+C-definable (G. 2010).

Definability

We know:

- The classes of planar graphs and K_5-minor free graphs are IFP-definable (G. 1998, 2008)
- For all k, the class of graphs of tree width at most k is IFP-definable (G., Mariño 1999)
- All nontrivial classes of graphs closed under taking minors are IFP+C-definable (G. 2010).

Conjecture

All nontrivial classes of graphs closed under taking minors are IFP*-definable.*

Classes Closed Under Induced Subgraphs

We know:

- IFP+C captures polynomial time on the class of interval graphs (Laubner 2010) and the class of chordal line graphs (G. 2010).

Classes Closed Under Induced Subgraphs

We know:

- IFP+C captures polynomial time on the class of interval graphs (Laubner 2010) and the class of chordal line graphs (G. 2010).
- IFP+C does not capture polynomial time on the class of rectangle intersection graphs, and if there is a logic that captures polynomial time on this class then there is a logic that captures polynomial time on all graphs.

Classes Closed Under Induced Subgraphs

We know:

- IFP+C captures polynomial time on the class of interval graphs (Laubner 2010) and the class of chordal line graphs (G. 2010).
- IFP+C does not capture polynomial time on the class of rectangle intersection graphs, and if there is a logic that captures polynomial time on this class then there is a logic that captures polynomial time on all graphs.

Open problem

Is there a logic that captures polynomial time on the class of unit square (or unit disk) intersection graphs? Maybe even IFP+C *?*

Classes Closed Under Induced Subgraphs

We know:

- IFP+C captures polynomial time on the class of interval graphs (Laubner 2010) and the class of chordal line graphs (G. 2010).
- IFP+C does not capture polynomial time on the class of rectangle intersection graphs, and if there is a logic that captures polynomial time on this class then there is a logic that captures polynomial time on all graphs.

Open problem

Is there a logic that captures polynomial time on the class of unit square (or unit disk) intersection graphs? Maybe even IFP+C *?*

Open problem

Is there a logic that captures polynomial time on classes of bounded rank width? Maybe even IFP+C *?*