Structure Theorem and Isomorphism Test for Graphs with Excluded Topological Subgraphs

Martin Grohe[1] Dániel Marx[2]

[1]Institut für Informatik
Humboldt-Universität zu Berlin, Germany

[2]Computer and Automation Research Institute,
Hungarian Academy of Sciences (MTA SZTAKI)
Budapest, Hungary,

STOC 2012
New York, NY
May 20, 2012

Overview

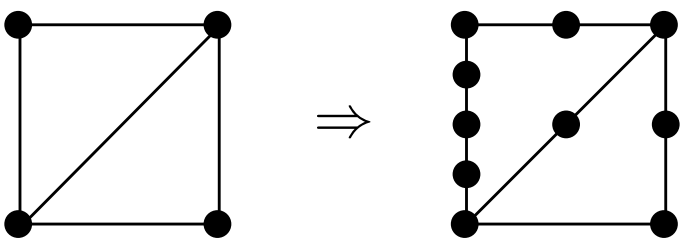
- Decomposition theorem for graphs excluding a topological minor (subdivision) of a fixed graph H.
- Algorithmic applications
  - Example: Partial Dominating Set
  - Isomorphism test.
- Warning: technical details and definitions are omitted.

Topological subgraphs

Definition

Subdivision of a graph: replacing each edge by a path of length 1 or more.

Graph H is a topological subgraph of G (or topological minor of G, or $H \leq_T G$) if a subdivision of H is a subgraph of G.

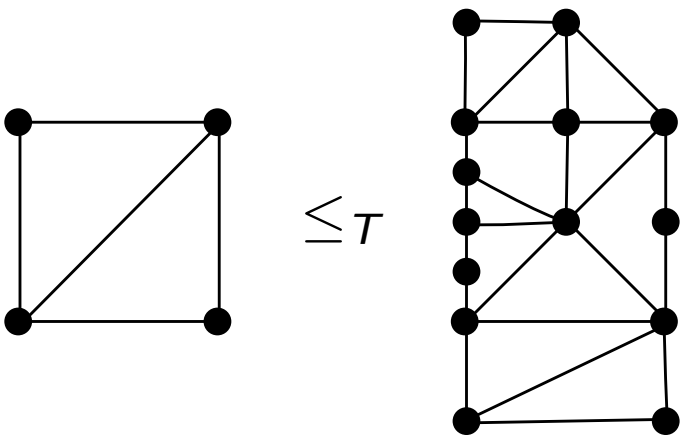


Topological subgraphs

Definition

Subdivision of a graph: replacing each edge by a path of length 1 or more.

Graph H is a topological subgraph of G (or topological minor of G, or $H \leq_T G$) if a subdivision of H is a subgraph of G.

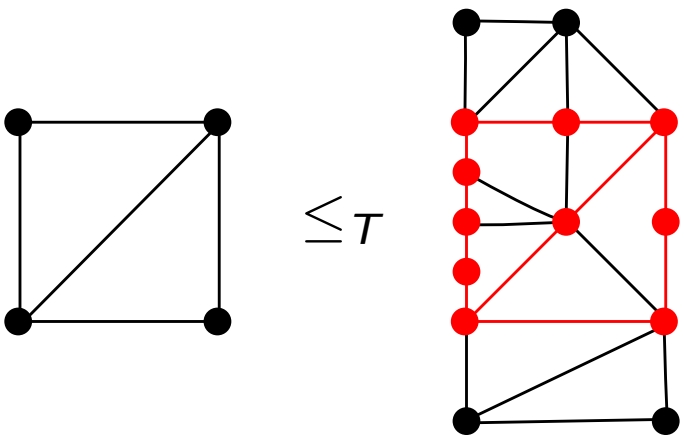


Topological subgraphs

Definition

Subdivision of a graph: replacing each edge by a path of length 1 or more.

Graph H is a topological subgraph of G (or topological minor of G, or $H \leq_T G$) if a subdivision of H is a subgraph of G.

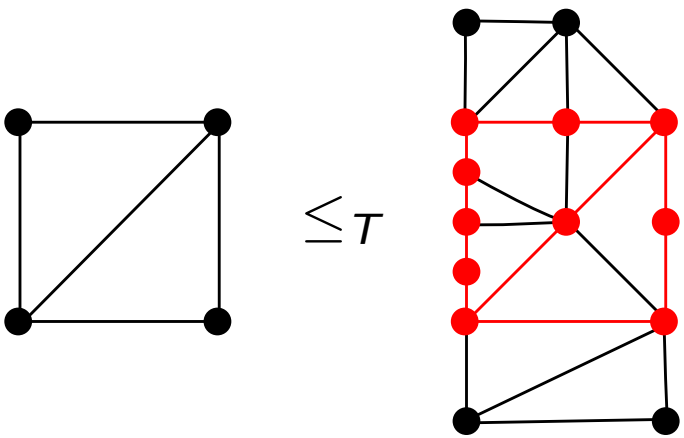


Topological subgraphs

Definition

Subdivision of a graph: replacing each edge by a path of length 1 or more.

Graph H is a topological subgraph of G (or topological minor of G, or $H \leq_T G$) if a subdivision of H is a subgraph of G.

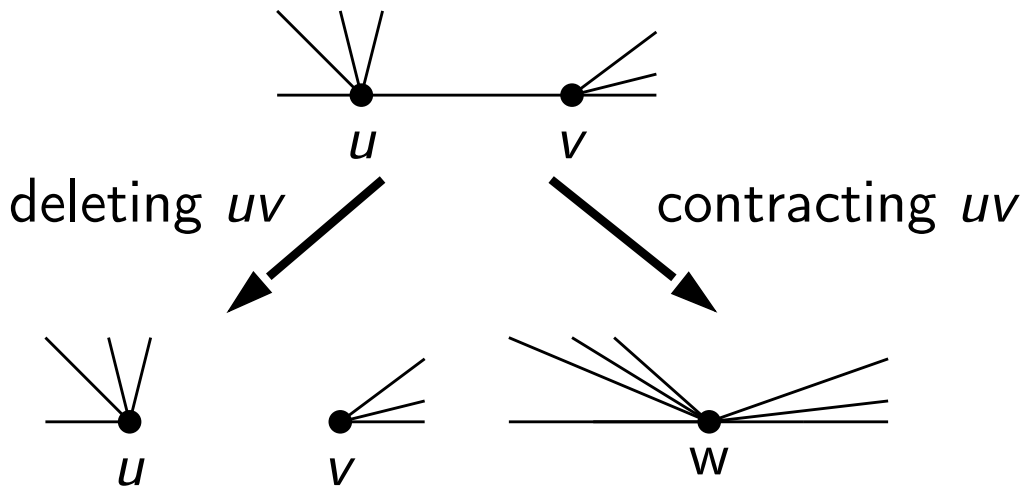


Equivalently, $H \leq_T G$ means that H can be obtained from G by removing vertices, removing edges, and dissolving degree two vertices.

Minors

Definition

Graph H is a minor G ($H \leq G$) if H can be obtained from G by deleting edges, deleting vertices, and contracting edges.



Note: $H \leq_T G \Rightarrow H \leq G$, but the converse is not necessarily true.

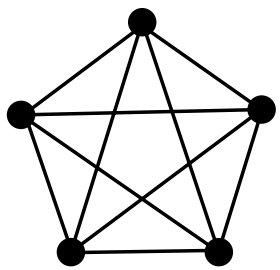
A classical result

Theorem [Kuratowski 1930]

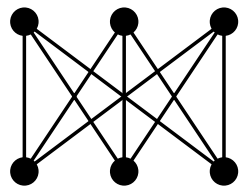
A graph G is planar if and only if $K_5 \not\leq_T G$ and $K_{3,3} \not\leq_T G$.

Theorem [Wagner 1937]

A graph G is planar if and only if $K_5 \not\leq G$ and $K_{3,3} \not\leq G$.



K_5



$K_{3,3}$

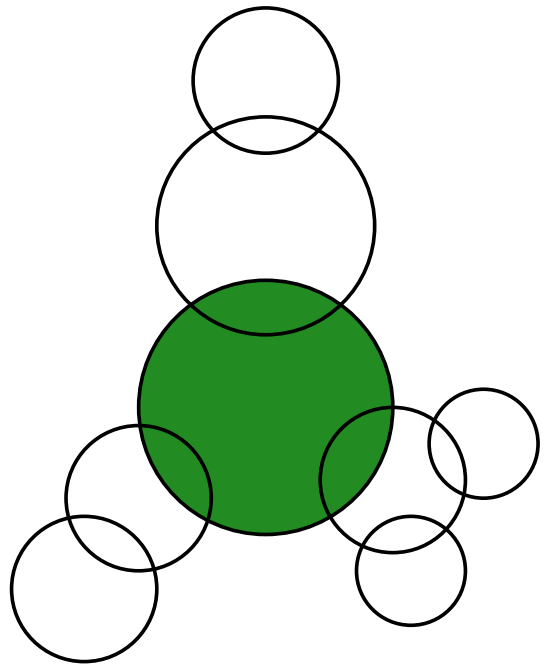
Structure theorem

Main question

- Can we say something about the structure of graphs **not** containing H as a minor?
  $\Rightarrow$ Work of Robertson and Seymour.
- Can we say something about the structure of graphs **not** containing H as a topological subgraph?
  $\Rightarrow$ This paper

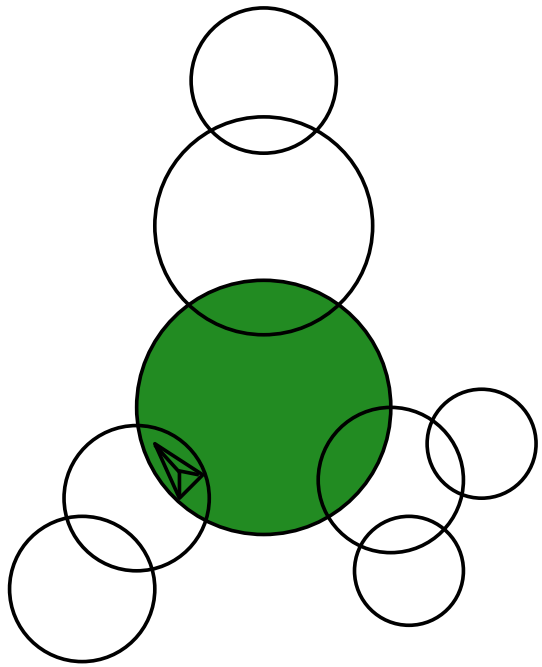
Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



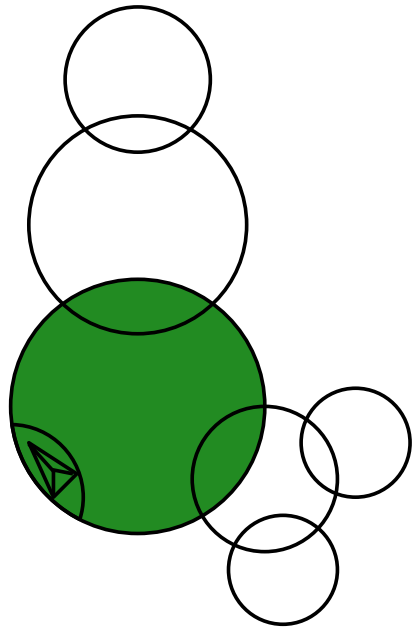
Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



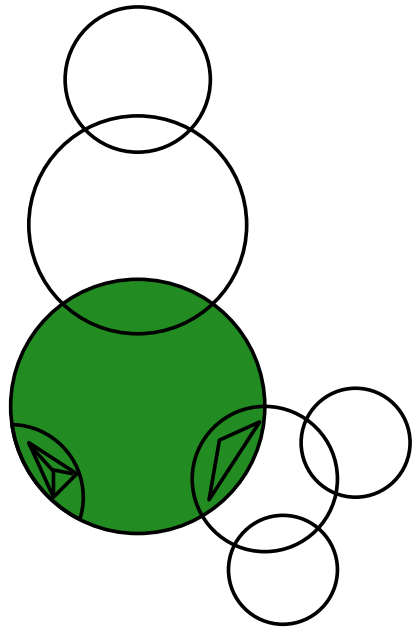
Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



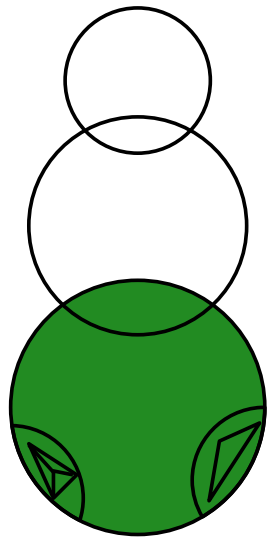
Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



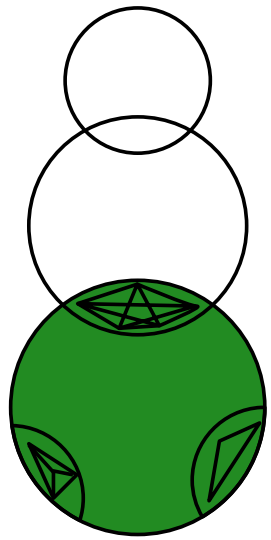
Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



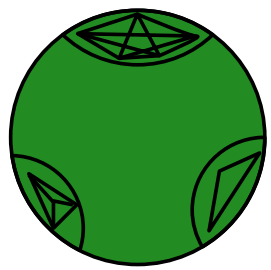
Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



Tree decompositions

Torso of a bag: we make the intersections with the adjacent bags cliques.



Structure theorems

Theorem [Robertson and Seymour]

Every H-minor free graph has a tree decomposition where the torso of every bag is "c_H-almost-embeddable."

Note: There is an $f(H) \cdot n^{O(1)}$ time algorithm for computing such a decomposition [Kawarabayashi-Wollan 2011].

Can we prove a similar result for the more general class of H-subdivision free graphs?

These classes are significantly more general: e.g., every 3-regular graph is K_5-subdivision free.

Structure theorems

Theorem [Robertson and Seymour]

Every H-minor free graph has a tree decomposition where the torso of every bag is "c_H-almost-embeddable."

Note: There is an $f(H) \cdot n^{O(1)}$ time algorithm for computing such a decomposition [Kawarabayashi-Wollan 2011].

Can we prove a similar result for the more general class of H-subdivision free graphs?

These classes are significantly more general: e.g., every 3-regular graph is K_5-subdivision free.

Structure theorems

New result

Every H-subdivision free graph has a tree decomposition where the torso of every bag is either

- K_{c_H}-minor free or
- has degree at most c_H with the exception of at most c_H vertices ("almost bounded degree").

Note: there is an $f(H) \cdot n^{O(1)}$ time algorithm for computing such a decomposition.

Structure theorems

New result

Every H-subdivision free graph has a tree decomposition where the torso of every bag is either

- "c_H-almost-embeddable" or
- has degree at most c_H with the exception of at most c_H vertices ("almost bounded degree").

Note: there is an $f(H) \cdot n^{O(1)}$ time algorithm for computing such a decomposition.

Proof overview

Star decomposition: tree decomposition where the tree is a star.

Local decomposition theorem

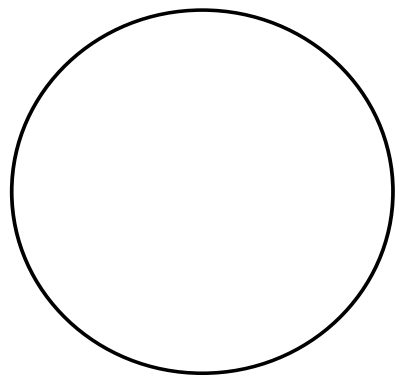
Given an H-subdivision free graph and a set S of at most a_H vertices, there is star decomposition where S is in the center bag and the torso of the center + (clique on S) either

(i) has bounded size.

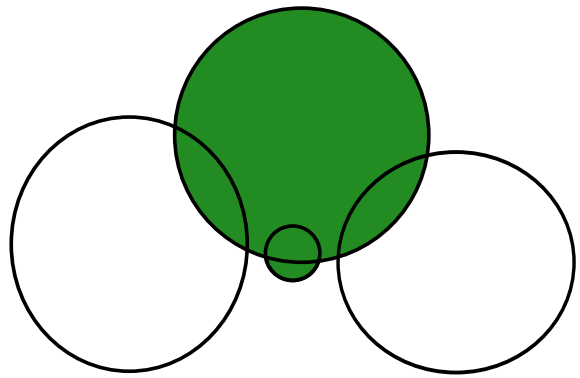
(ii) excludes a clique minor.

(iii) has almost-bounded degree.

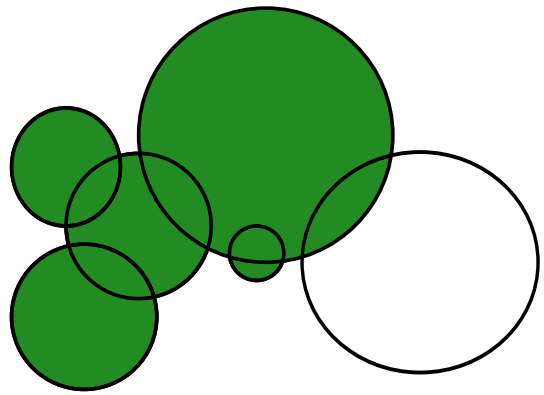
Iterating local decompositions



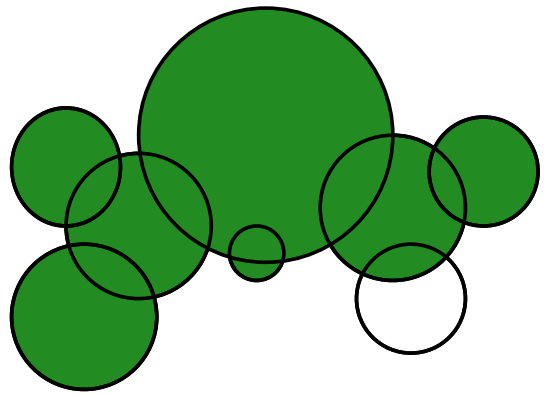
Iterating local decompositions



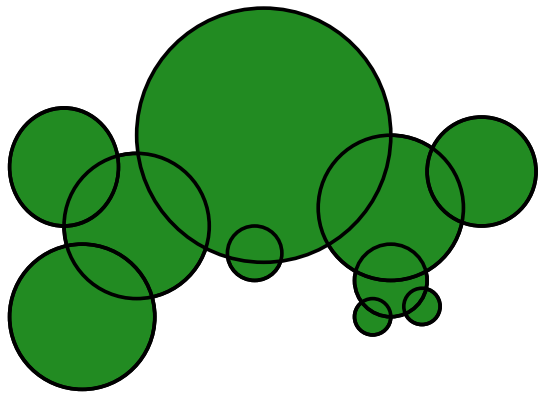
Iterating local decompositions



Iterating local decompositions



Iterating local decompositions



Algorithmic applications

New result

Every H-subdivision free graph has a tree decomposition where the torso of every bag is either

- "c_H-almost-embeddable" or
- has degree at most c_H with the exception of at most c_H vertices ("almost bounded degree").

General message:

If a problem can be solved both

- on (almost-) embeddable graphs,
- on (almost-) bounded degree graphs and

then these results can be raised to

- H-subdivision free graphs

without too much extra effort.

Partial Dominating Set

Partial Dominating Set

Input: graph G, integer k
Find: a set S of at most k vertices whose closed neighborhood has maximum size

Theorem

Partial Dominating Set can be solved in time $f(H, k) \cdot n^{O(1)}$ on H-subdivision free graphs.

Graph Isomorphism

Graph Isomorphism

Input: Graphs G_1 and G_2
Decide: Are G_1 and G_2 isomorphic?

Not known to be polynomial-time solvable, not believed to be NP-hard.

Related problems:

- Decide if two graphs are isomorphic.
- Find an isomorphism.
- Compute a canonical label for the graph.
- Compute a canonical labeling of the vertices.

Graph Isomorphism

Graph Isomorphism

Input: Graphs G_1 and G_2
Decide: Are G_1 and G_2 isomorphic?

Not known to be polynomial-time solvable, not believed to be NP-hard.

Related problems:

- Decide if two graphs are isomorphic.
- Find an isomorphism.
- Compute a canonical label for the graph.
- Compute a canonical labeling of the vertices.

Graph Isomorphism

Theorem [Luks 1982] [Babai, Luks 1983]

For every fixed d, Graph Isomorphism can be solved in polynomial time on graphs with maximum degree d.

Theorem [Ponomarenko 1988]

For every fixed H, Graph Isomorphism can be solved in polynomial time on H-minor free graphs.

New result

For every fixed H, Graph Isomorphism can be solved in polynomial-time on H-subdivision free graphs.

Note: running time is $n^{f(H)}$, not FPT parameterized by H.

Good news

The paper contains no algebra or group theory.

Good news

The paper contains no algebra or group theory.

Bad news

The paper contains no algebra or group theory.

Graph Isomorphism

New result

For every fixed H, Graph Isomorphism can be solved in polynomial-time on H-subdivision free graphs.

Proof idea:

- Compute a tree decomposition for the graph.
- Use bottom up dynamic programing to compute a canonical label for every subtree.
- We can compute a canonical label for each torso using the bounded-degree or the excluded minor algorithm.
- Incorporate the labels of the children as annotation.

Graph Isomorphism

Huge problem

Even if G_1 and G_2 are isomorphic, we are not guaranteed to obtain isomorphic tree decompositions.

Idea 1:

Try to make the algorithm invariant (avoid arbitrary choices in the algorithms). Not known how to do this already for bounded-treewidth graphs.

Idea 2:

Use the more general notion of treelike decompositions and try to find such decompositions in an invariant way.

Graph Isomorphism

Huge problem

Even if G_1 and G_2 are isomorphic, we are not guaranteed to obtain isomorphic tree decompositions.

Idea 1:

Try to make the algorithm invariant (avoid arbitrary choices in the algorithms). Not known how to do this already for bounded-treewidth graphs.

Idea 2:

Use the more general notion of treelike decompositions and try to find such decompositions in an invariant way.

Graph Isomorphism

Huge problem

Even if G_1 and G_2 are isomorphic, we are not guaranteed to obtain isomorphic tree decompositions.

Idea 1:

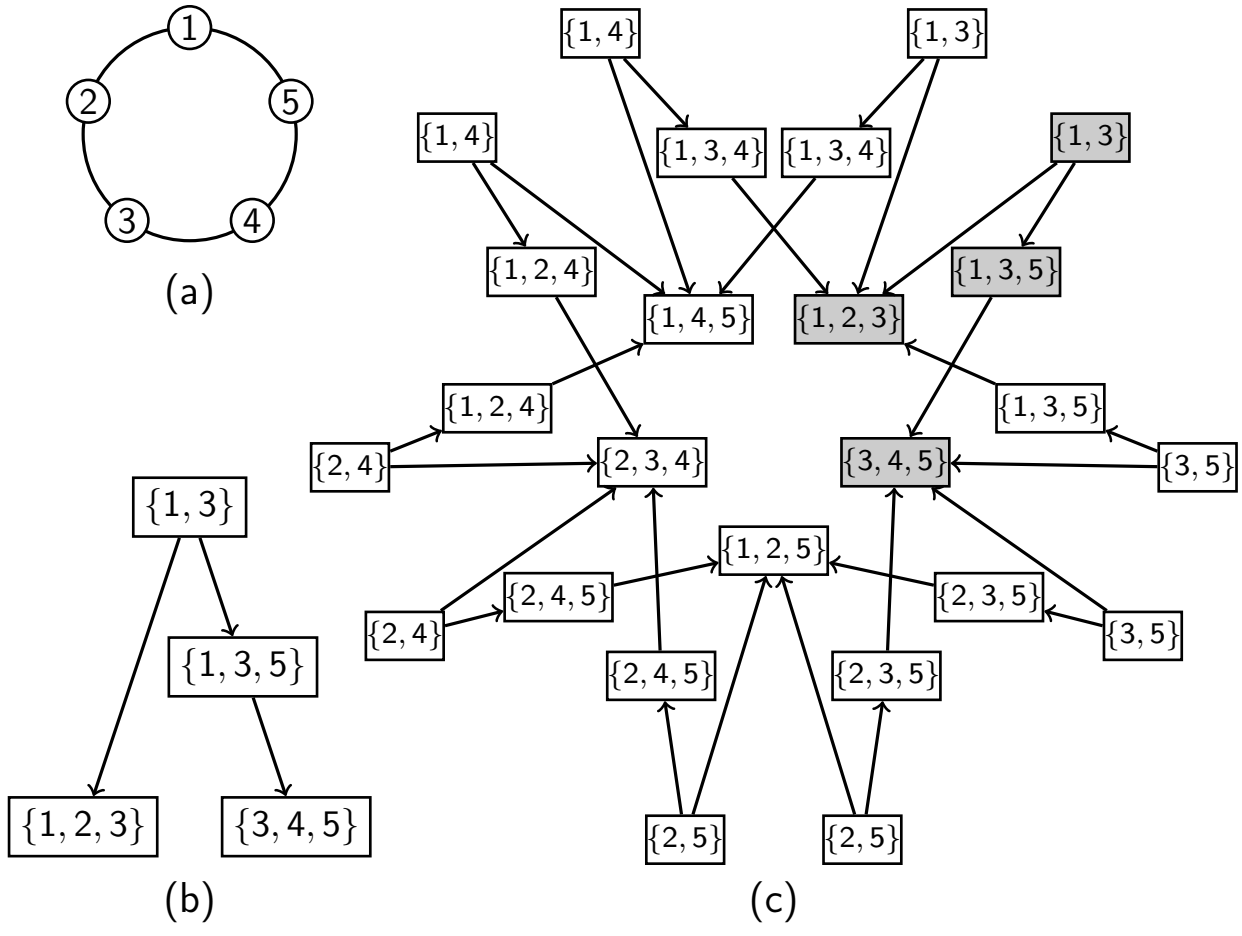
Try to make the algorithm invariant (avoid arbitrary choices in the algorithms). Not known how to do this already for bounded-treewidth graphs.

Idea 2:

Use the more general notion of treelike decompositions and try to find such decompositions in an invariant way.

Treelike decompositions

[Grohe 2008] generalized the notion of tree decompositions to acyclic treelike decompositions:



Graph Isomorphism

New result

Every H-subdivision free graph has a tree decomposition where the torso of every bag is either

- "c_H-almost-embeddable" or
- has degree at most c_H with the exception of at most c_H vertices ("almost bounded degree").

Theorem

We can compute such a treelike decomposition in time $n^{f(H)}$ such that for isomorphic graphs we create isomorphic decompositions.

Now the difficulty disappears: we can compute a canonical label with a bottom-up dynamic programming approach.

Summary

- Structure theorem for decomposing H-subdivision free graphs into almost-embeddable and almost bounded-degree graphs.
- Algorithmic applications on H-subdivision free graphs:
  - $f(k, H) \cdot n^{O(1)}$ time algorithm for Partial Dominating Set.
  - $n^{f(H)}$ time algorithm for Graph Isomorphism.