Sparsification Upper and Lower Bounds for Graph Problems and Not-All-Equal SAT

Bart M.P. Jansen and Astrid Pieterse

Technische Universiteit Eindhoven
University of Technology

September 16, 2015

Sparsification

Kernelization

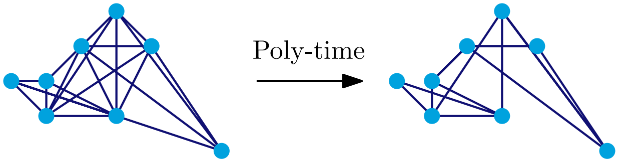
In polynomial time reduce the size of an instance

- Size depends only on the parameter

Sparsification

In polynomial time, create an instance that is less dense

- Has sub-quadratic number of edges
- Use kernelization framework
- Parameter: $n = |V|$



Background

Graph problems: trivial kernel of size $O(n^2)$. Can we do better?
Use generalized kernel for lower bounds

Unless $NP \subseteq coNP/poly$:

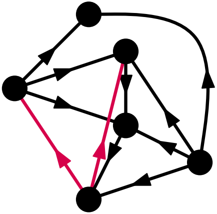
- VERTEX COVER, no generalized kernel of size $O(n^{2-\varepsilon})$
  d-CNF-SAT, no generalized kernel of size $O(n^{d-\varepsilon})$
  - Dell, van Melkebeek (J ACM14)
- TREEWIDTH, no generalized kernel of size $O(n^{2-\varepsilon})$
  - Jansen (IPEC13)

Do other graph and logic problems allow polynomial time sparsification?

Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $NP \subseteq coNP/poly$.

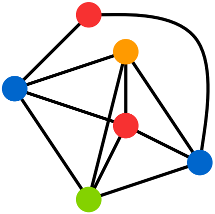
- FEEDBACK ARC SET



Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $NP \subseteq coNP/poly$.

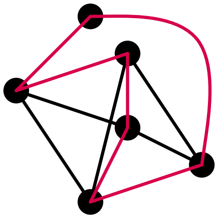
- FEEDBACK ARC SET
- 4-COLORING
  - And thereby k-COLORING for $k \geqslant 4$



Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $NP \subseteq coNP/poly$.

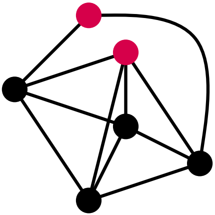
- FEEDBACK ARC SET
- 4-COLORING
  - And thereby k-COLORING for $k \geqslant 4$
- HAMILTONIAN CYCLE



Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

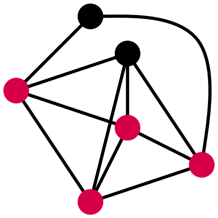
- FEEDBACK ARC SET
- 4-COLORING
  - And thereby k-COLORING for $k \geqslant 4$
- HAMILTONIAN CYCLE
- DOMINATING SET



Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $NP \subseteq coNP/poly$.

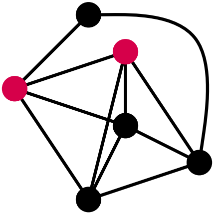
- Feedback Arc Set
- 4-Coloring
  - And thereby k-Coloring for $k \geqslant 4$
- Hamiltonian Cycle
- Dominating Set
- Non Blocker
  - Kernel with $O(k)$ vertices by Dehne et al. (SOFSEM06)



Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $NP \subseteq coNP/poly$.

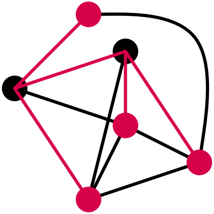
- Feedback Arc Set
- 4-Coloring
  - And thereby k-Coloring for $k \geqslant 4$
- Hamiltonian Cycle
- Dominating Set
- Non Blocker
  - Kernel with $O(k)$ vertices by Dehne et al. (SOFSEM06)
- Connected Dominating Set



Our results

Problems without a generalized kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $NP \subseteq coNP/poly$.

- FEEDBACK ARC SET
- 4-COLORING
  - And thereby k-COLORING for $k \geqslant 4$
- HAMILTONIAN CYCLE
- DOMINATING SET
- NON BLOCKER
  - Kernel with $O(k)$ vertices by Dehne et al. (SOFSEM06)
- CONNECTED DOMINATING SET
- MAXIMUM LEAF SPANNING TREE
  - Kernel with $O(k)$ vertices by Estivill-Castro et al. (ACiD05)



Our results

(Non)-sparsifiability of CNF-formulas

- d-NAE-SAT has a generalized kernel of size $O(n^{d-1})$
  - Matches known lower bound

Our results

(Non)-sparsifiability of CNF-formulas

- d-NAE-SAT has a generalized kernel of size $O(n^{d-1})$
  - Matches known lower bound

Compare to d-CNF-SAT, $O(n^{d-\varepsilon})$ kernel unlikely

Not-All-Equal satisfiability

d-NAE-SAT

- Input: CNF-formula $\mathcal{F}$, each clause contains at most d literals.

$$\mathcal{F} = \underbrace{(x \vee \neg y \vee \ldots \vee z)}_{\text{Clause, } \leqslant d \text{ literals}} \wedge (\neg x \vee \neg z \vee \ldots \vee \neg y) \wedge \ldots$$

- Parameter: The number of variables n.
- Question: Find a truth assignment such that every clause contains a true and a false literal?

Not-All-Equal Satisfiability: kernel bounds

No generalized kernel of size $O(n^{d-1-\varepsilon})$, unless $NP \subseteq coNP/poly$.

- Linear parameter transformation d-CNF-SAT to $(d+1)$-NAE-SAT
  - Shown by Jansen et al. (InformComput13)
- Is this a tight lower bound?
  - Not trivial.
- Provide a generalized kernel via d-HYPERGRAPH 2-COLORABILITY

$$\underbrace{(x, k)}_{d\text{-NAE-Sat}} \xrightarrow{\text{lpt}} \underbrace{(x', k')}_{d\text{-Hypergraph } 2\text{-Colorability}} \xrightarrow{\text{kernel}} \underbrace{(x'', k'')}_{d\text{-Hypergraph } 2\text{-Colorability}}$$

d-Hypergraph 2-Colorability

d-HYPERGRAPH 2-COLORABILITY

- Input: Hypergraph, every edge contains at most d vertices.
- Parameter: The number of vertices n.
- Question: Color each vertex with red/blue such that every edge contains a red and a blue vertex?

Kernel

d-HYPERGRAPH 2-COLORABILITY has a kernel with $O(n^{d-1})$ edges.

d-Hypergraph 2-Colorability: Kernel

Construction

- Let every edge have exactly d vertices
- Edges $e_1, \ldots, e_m$
- Enumerate all size $d - 1$ subsets of V as $S_1, S_2, \ldots, S_\ell$
- Create $(0, 1)$-matrix M

$$\begin{array}{c} \\ S_1 \\ S_2 \\ \ldots \\ S_\ell \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & \ldots & e_m \end{array} \\ \begin{pmatrix} S_1 \subseteq e_1 & S_1 \subseteq e_2 & \ldots & S_1 \subseteq e_m \\ S_2 \subseteq e_1 & S_2 \subseteq e_2 & \ldots & S_2 \subseteq e_m \\ \ldots & \ldots & \ldots & \ldots \\ S_\ell \subseteq e_1 & S_\ell \subseteq e_2 & \ldots & S_\ell \subseteq e_m \end{pmatrix} \end{array}$$

- Compute a base of the columns of this matrix.
- This results in a subset of the edges of G

d-Hypergraph 2-Colorability: Kernel

Size

- Matrix M has at most $\binom{n}{d-1} \leqslant n^{d-1}$ rows
- Any base of M contains at most n^{d-1} edges

Correctness

- If G is 2-colorable, so is the reduced graph
- Edges in the base are split implies removed edges split
  - Proof uses a lemma by Lovász (1976)
  - Critical 3-chromatic hypergraphs have at most $O(n^{d-1})$ edges
  - Transferred proof idea into kernel

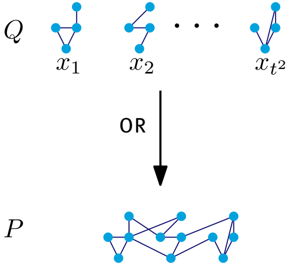
Degree-2 cross-composition

Degree-2 cross-composition to P gives kernel lower bound $O(n^2)$

Degree-2 cross-composition

By Bodlaender et al. (SIDMA14)

- Start from any NP-hard problem Q
- Give a polynomial time algorithm
- Input: t^2 similar instances of Q
- Output: An instance (y, k) of P, where
  - $k = O(t \cdot \max |x_i|^c)$
  - (y, k) is a logical OR of the inputs



4-Coloring

Goal

Prove that there is no kernel of size $O(n^{2-\varepsilon})$ for 4-COLORING.

4-Coloring

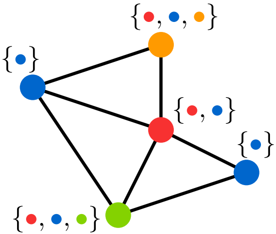
Goal

Prove that there is no kernel of size $O(n^{2-\varepsilon})$ for 4-COLORING.

4-LIST COLORING

We use list coloring with 4 colors in total, instead of 4-COLORING.

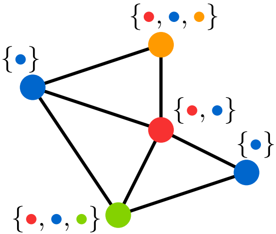
- Vertex has list of allowed colors
    - Subset of {●, ●, ●, ●}



4-Coloring

Goal

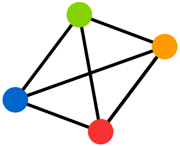
Prove that there is no kernel of size $O(n^{2-\varepsilon})$ for 4-COLORING.



4-LIST COLORING

We use list coloring with 4 colors in total, instead of 4-COLORING.

- Vertex has list of allowed colors
  - Subset of $\{\bullet, \bullet, \bullet, \bullet\}$
- Transform back to 4-COLORING



4-Coloring

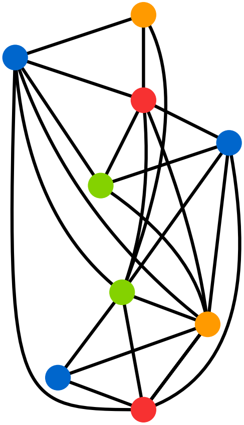
Goal

Prove that there is no kernel of size $O(n^{2-\varepsilon})$ for 4-Coloring.

4-List Coloring

We use list coloring with 4 colors in total, instead of 4-Coloring.

- Vertex has list of allowed colors
  - Subset of {●, ●, ●, ●}
- Transform back to 4-Coloring

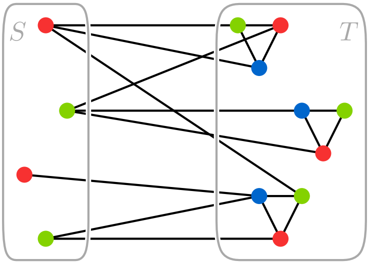


4-Coloring: Cross-composition

NP-hard starting problem

2-3-COLORING ON TRIANGLE-SPLIT GRAPHS

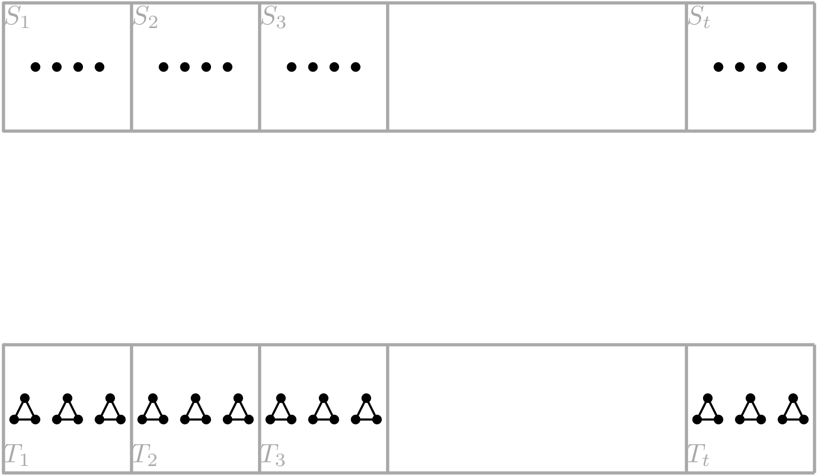
- Input: Graph $G = (S \cup T, E)$ where S is an independent set and T consists of disjoint triangles.
- Question: Does G have a proper 3-coloring, such that S is colored using only 2 colors?
  - We call this a 2-3-*coloring* of G.



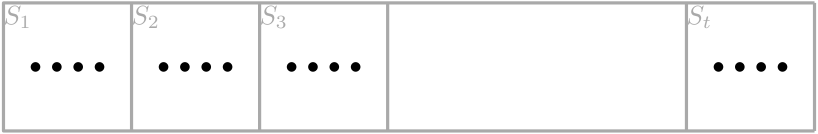
4-Coloring: Cross-composition

- Assume we have t^2 instances
- Let $|S| = n$ and $|T| = 3m$ for all instances
- Construct a cross-composition
  - At most $O(t \cdot (n + m))$ vertices
- We cannot copy all vertices
  - But we can keep all edges
  - Method introduced by Dell and Marx (SODA12)
- Enumerate instances as X_{ij}, where $i = 1, \ldots, t$

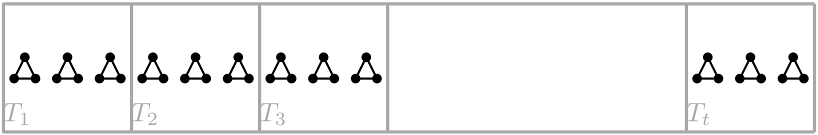
4-Coloring: First idea



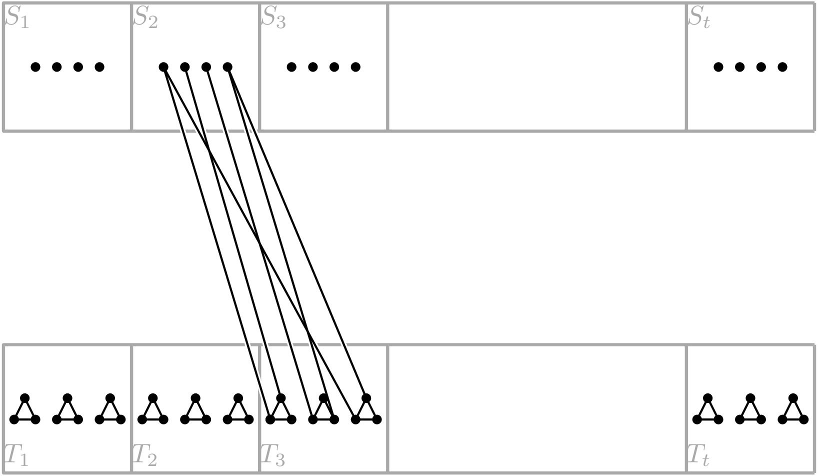
4-Coloring: First idea



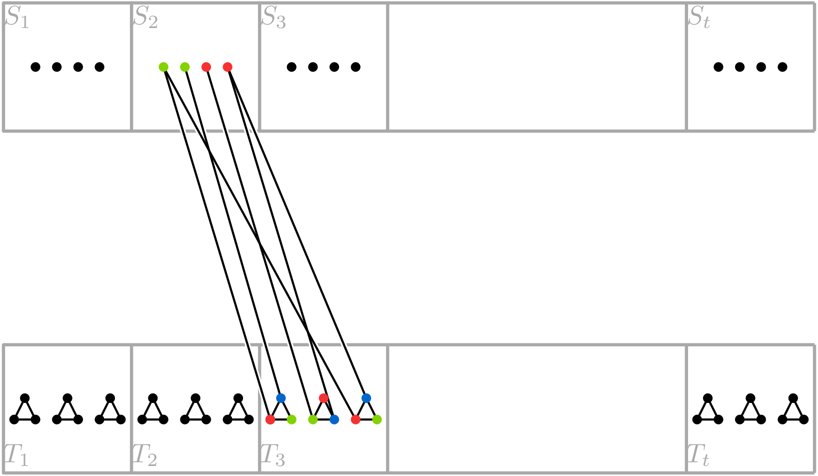
Instance X_{23}?



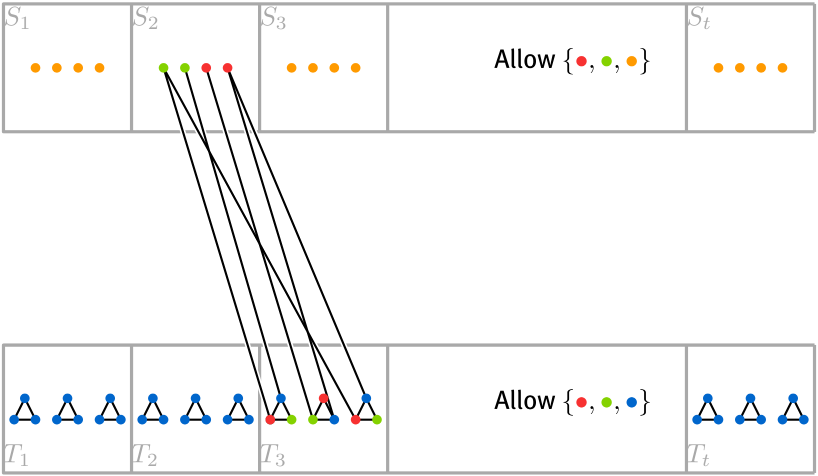
4-Coloring: First idea



4-Coloring: First idea

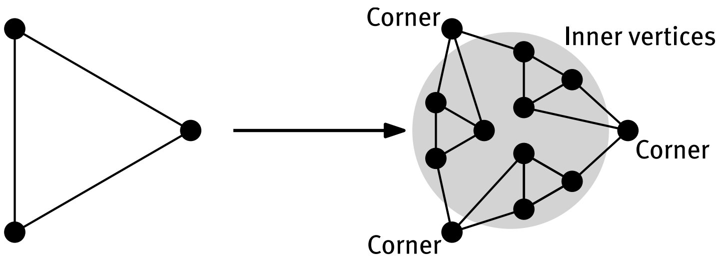


4-Coloring: First idea



4-Coloring: Triangular gadget

This is not a valid coloring of the triangles. Replace them:

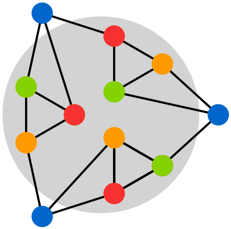


Allow the new vertices to be red, green, orange, or blue.

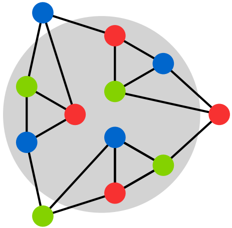
4-Coloring: Triangular gadget

Useful properties

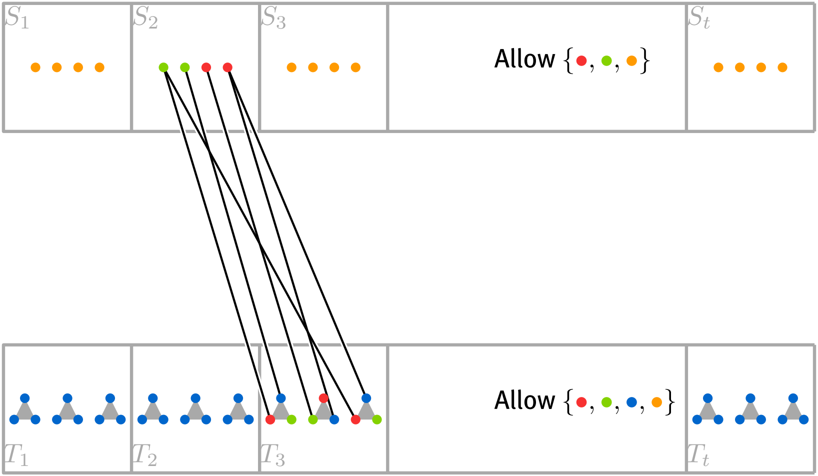
We can color all corners blue



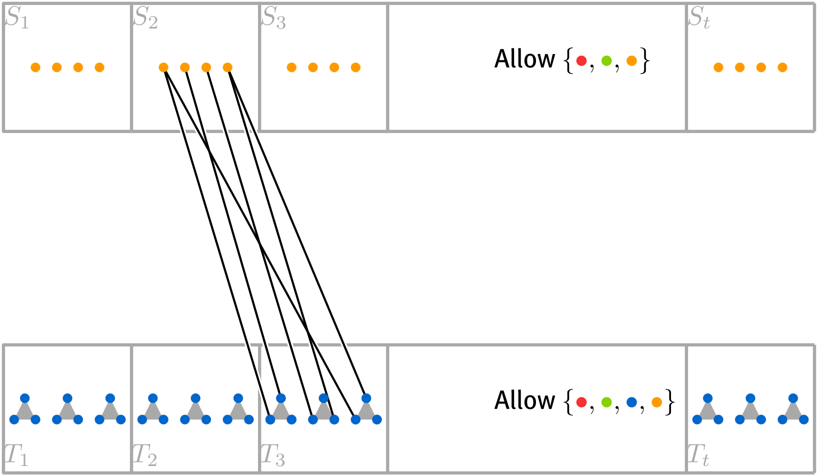
If 3-colored: ordinary triangle



4-Coloring: Improved



4-Coloring: Improved

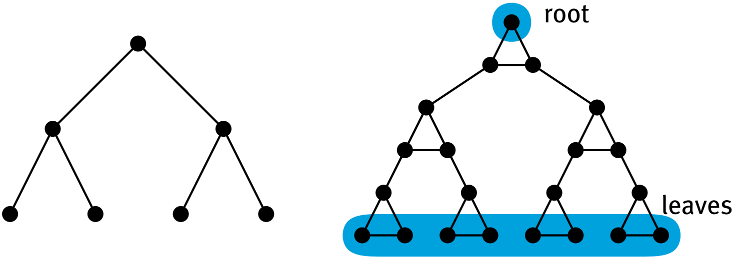


4-Coloring: Treegadgets

Ensure one S_i and one T_j are colored without orange

Treegadget

Complete binary tree where every vertex is replaced by a triangle:

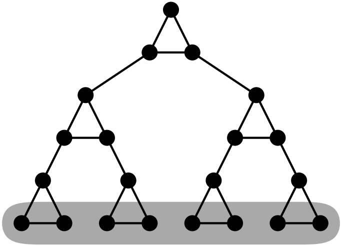


4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 1

If none of the leaves is colored orange, the root must be orange.

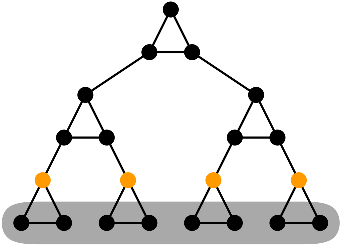


4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 1

If none of the leaves is colored orange, the root must be orange.

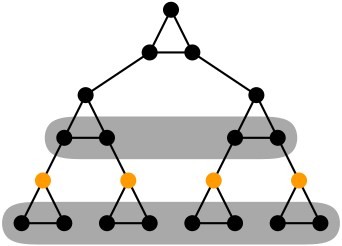


4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 1

If none of the leaves is colored orange, the root must be orange.

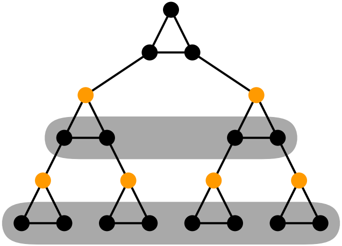


4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 1

If none of the leaves is colored orange, the root must be orange.

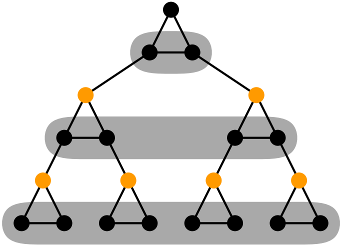


4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 1

If none of the leaves is colored orange, the root must be orange.

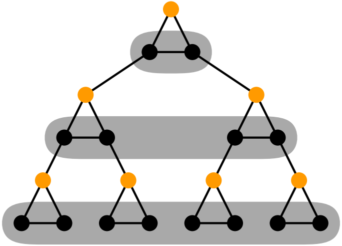


4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 1

If none of the leaves is colored orange, the root must be orange.



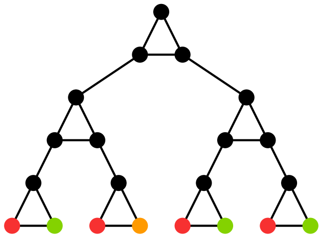
4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 2

We can extend a coloring of the leaves, to color the entire tree.

- If at least one of the leaves is orange, the root can be red or green.



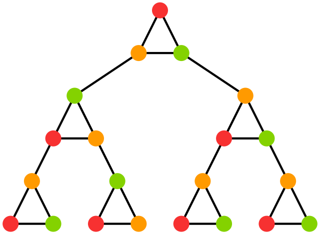
4-Coloring: Treegadgets

Suppose we want to 3-color a gadget, using red, green and orange.

Property 2

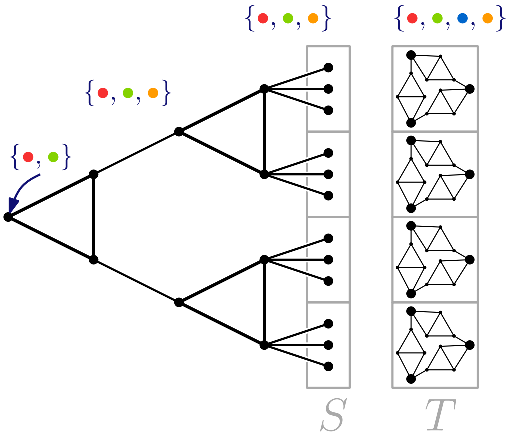
We can extend a coloring of the leaves, to color the entire tree.

- If at least one of the leaves is orange, the root can be red or green.



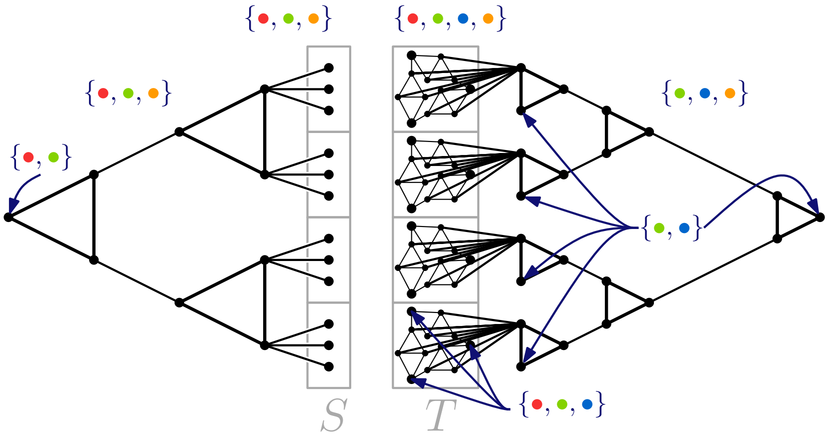
4-Coloring: Cross-composition

Ensure one group in S is colored with red and green.



4-Coloring: Cross-composition

Ensure one group in T is colored with red, green and orange.



4-Coloring: Cross-composition

- ✓ The number of vertices of G is allowed: $O(t \cdot \max |X_{ij}|)$.
- ✓ Can be done in polynomial time.
- ✓ If some X_{ij} is 2-3-colorable, G is 4-colorable.
- ✓ If G is 4-colorable, there exists an X_{ij} that is 2-3-colorable.

Theorem

4-Coloring does not have a generalized kernel of size $O(n^{2-\varepsilon})$ for $\varepsilon > 0$, unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

Conclusion

- Non-sparsifiability of several classic graph problems
  - Implications for k-NON BLOCKER and k-MAX LEAF SPANNING TREE
- Generalized kernel for NAE-SAT
  - One of the first non-trivial sparsifications.

Open problems

- Does 3-COLORING allow sparsification?
- Sparsification of NP-hard problem on general graphs
- Sparsification of edge-based problems
  - Max Cut

Conclusion

- Non-sparsifiability of several classic graph problems
  - Implications for k-NON BLOCKER and k-MAX LEAF SPANNING TREE
- Generalized kernel for NAE-SAT
  - One of the first non-trivial sparsifications.

Open problems

- Does 3-COLORING allow sparsification?
- Sparsification of NP-hard problem on general graphs
- Sparsification of edge-based problems
  - Max Cut

Thank you for your attention