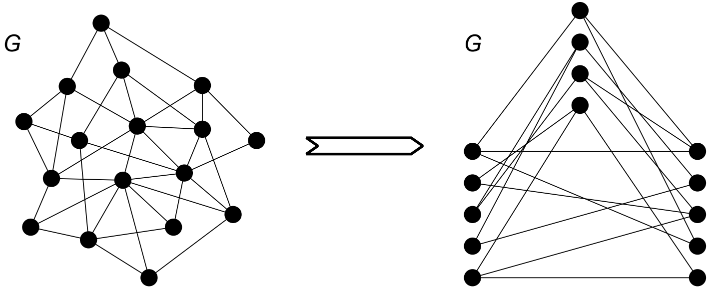
Data Reduction and Combinatorics of the 3-Colorability Problem

Oliver Schaudt

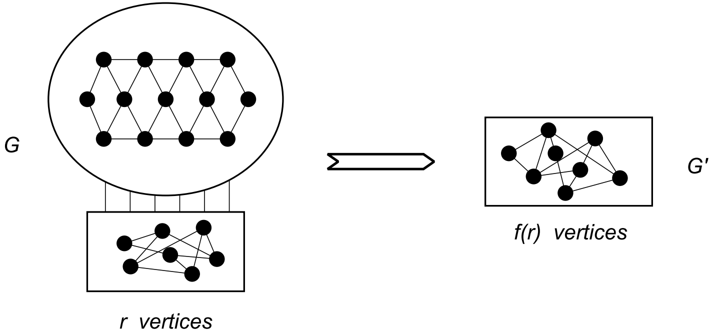
RWTH Aachen University

with Maria Chudnovsky, Jan Goedgebeur, and Mingxian Zhong

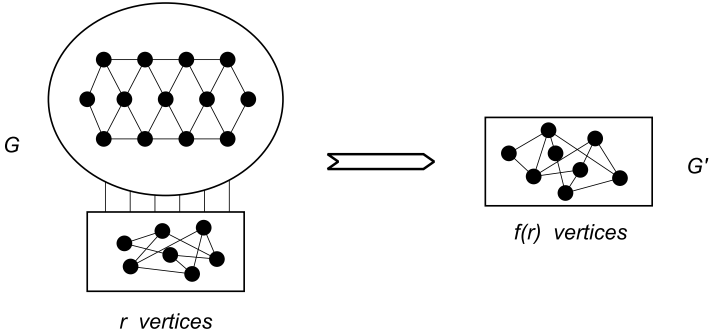
k-Colorability



Kernelization

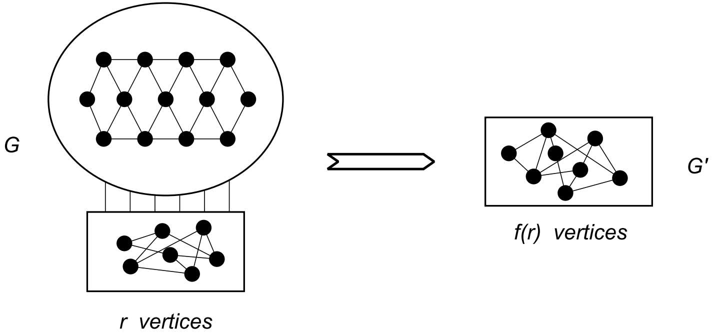


Kernelization



- assume that G can be made *nice* by removing r vertices

Kernelization



- assume that G can be made *nice* by removing r vertices
- a **kernelization** is a polynomial time algorithm that computes an equivalent instance G' of order $f(r)$

Kernelization

Problem

What is a *nice* graph property, meaning it allows for such a kernelization?

Kernelization

Problem

What is a *nice* graph property, meaning it allows for such a kernelization?

Answer

Every graph class in which we can solve the **list k-colorability problem** in polynomial time: every vertex is equipped with a list $L(v) \subseteq \{1, \ldots, k\}$ of admissible colors.

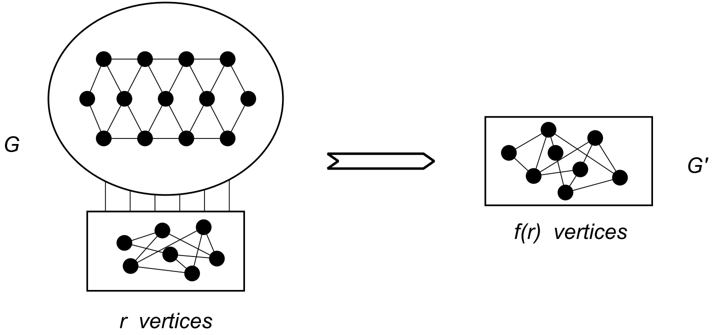
Kernelization

Problem
What is a *nice* graph property, meaning it allows for such a kernelization?

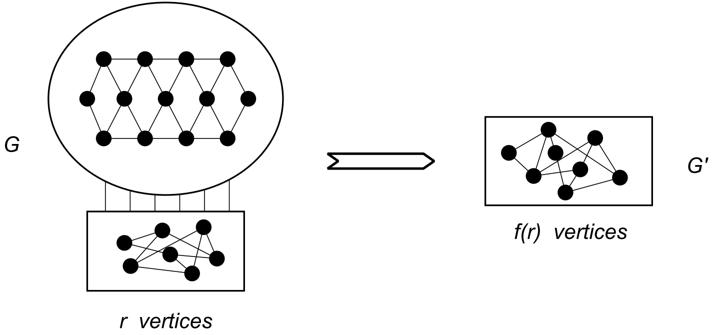
Answer
Every graph class in which we can solve the **list k-colorability problem** in polynomial time: every vertex is equipped with a list $L(v) \subseteq \{1, \ldots, k\}$ of admissible colors.

Theorem (Bonomo, Chudnovsky, Maceli, S, Stein, Zhong '14)
The list 3-colorability problem can be solved in polynomial time for P_7-free graphs.

Polynomial kernelization



Polynomial kernelization

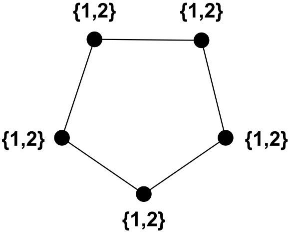


Problem

When is there a **polynomial kernelization**, meaning $f(r)$ is of order $r^{O(1)}$?

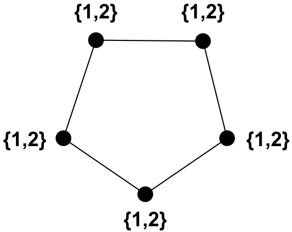
Polynomial kernelization

- let G be a graph and let $L(v) \subseteq \{1, \ldots, k\}$ for every vertex v



Polynomial kernelization

- let G be a graph and let $L(v) \subseteq \{1, \ldots, k\}$ for every vertex v



- we call (G, L) a **k-obstruction** if
  (a) there is no coloring for (G, L), and
  (b) there is a coloring if we remove any vertex from G

Polynomial kernelization

Theorem (Jansen & Kratsch 2013)

Let $\mathcal{P}$ be a graph property closed under deleting vertices. Assume there are only finitely many k-obstructions with property $\mathcal{P}$.

Polynomial kernelization

Theorem (Jansen & Kratsch 2013)

Let $\mathcal{P}$ be a graph property closed under deleting vertices. Assume there are only finitely many k-obstructions with property $\mathcal{P}$. There is a polynomial kernel of order $r^{O(1)}$ for the k-colorability problem on the graphs that have property $\mathcal{P}$ after removing r vertices.

Polynomial kernelization

Theorem (Jansen & Kratsch 2013)

Let $\mathcal{P}$ be a graph property closed under deleting vertices. Assume there are only finitely many k-obstructions with property $\mathcal{P}$. There is a polynomial kernel of order $r^{O(1)}$ for the k-colorability problem on the graphs that have property $\mathcal{P}$ after removing r vertices.

Theorem (Jansen & Kratsch 2013)

There are only finitely many P_4-free k-obstructions, for all k.

Polynomial kernelization

Theorem (Jansen & Kratsch 2013)

Let $\mathcal{P}$ be a graph property closed under deleting vertices. Assume there are only finitely many k-obstructions with property $\mathcal{P}$. There is a polynomial kernel of order $r^{O(1)}$ for the k-colorability problem on the graphs that have property $\mathcal{P}$ after removing r vertices.

Theorem (Jansen & Kratsch 2013)

There are only finitely many P_4-free k-obstructions, for all k.

If a property applies to all paths, there is no such polynomial kernelization.

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

Let H be a connected graph. The following two assertions are equivalent.

- *The 3-colorability problem admits a polynomial kernel of order $r^{O(1)}$ on the graphs that are H-free after removing r vertices.*
- *H is a path on at most six vertices.*

We assume that $NP \not\subseteq coNP/poly$.

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

Let H be a connected graph. The following two assertions are equivalent.

- *The 3-colorability problem admits a polynomial kernel of order $r^{O(1)}$ on the graphs that are H-free after removing r vertices.*
- *H is a path on at most six vertices.*

We assume that $NP \not\subseteq coNP/poly$.

Open Problem

Characterize the existence of a polynomial kernel for arbitrary graph properties closed under vertex deletion.

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

There are only finitely many P_6-free 3-obstructions.

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

There are only finitely many P_6-free 3-obstructions.

You cannot do better than P_6.

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

There are only finitely many P_6-free 3-obstructions.

You cannot do better than P_6.

- we do know these obstructions have at most $2^{2^{104}}$ vertices

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

There are only finitely many P_6-free 3-obstructions.

You cannot do better than P_6.

- we do know these obstructions have at most $2^{2^{104}}$ vertices
- gives a kernel of order $r^{3 \cdot 2^{2^{104}}}$

Polynomial kernelization

Theorem (Chudnovsky, Goedgebeur, S, Zhong 2016)

There are only finitely many P_6-free 3-obstructions.

You cannot do better than P_6.

- we do know these obstructions have at most $2^{2^{104}}$ vertices
- gives a kernel of order $r^{3\cdot 2^{2^{104}}}$
- there are 1 441 407 obstructions on at most 9 vertices

Easy case: lists have size at most two

Easy case: lists have size at most two

- pick a vertex v in G

Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:

Easy case: lists have size at most two

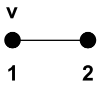
- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:

Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:

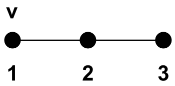
Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:



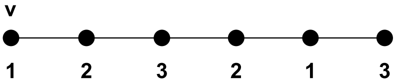
Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:



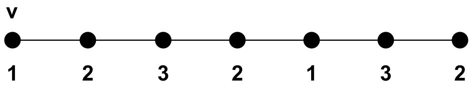
Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:



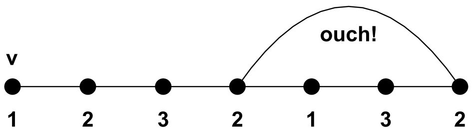
Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:



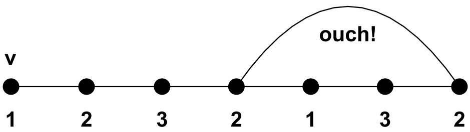
Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:



Easy case: lists have size at most two

- pick a vertex v in G
- if we give v a feasible color, this coloring propagates along a path until a contradiction appears:



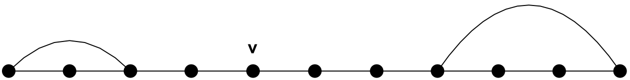
- doing this for both colors on v's list, we see that G is the union of two paths (not disjoint) starting in v

Easy case: lists have size at most two

- G is the union of two paths (not disjoint) starting in a common vertex

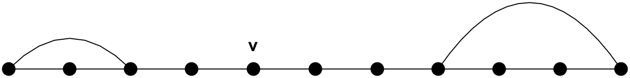
Easy case: lists have size at most two

- G is the union of two paths (not disjoint) starting in a common vertex



Easy case: lists have size at most two

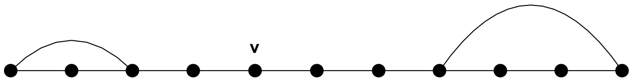
- G is the union of two paths (not disjoint) starting in a common vertex



- using a computer search, we can prove that both paths have bounded length

Easy case: lists have size at most two

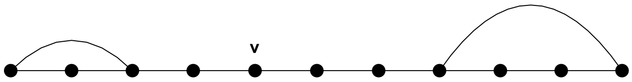
- G is the union of two paths (not disjoint) starting in a common vertex



- using a computer search, we can prove that both paths have bounded length
- thus the number of such obstructions is finite

Easy case: lists have size at most two

- G is the union of two paths (not disjoint) starting in a common vertex



- using a computer search, we can prove that both paths have bounded length
- thus the number of such obstructions is finite

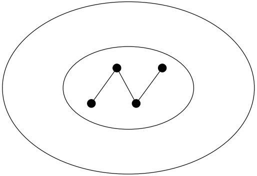
Thanks!

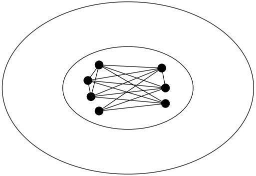
Obstructions with arbitrary lists

- our P_6-free obstruction has a connected dominating set inducing either a P_4 or a P_4-free graph

Obstructions with arbitrary lists

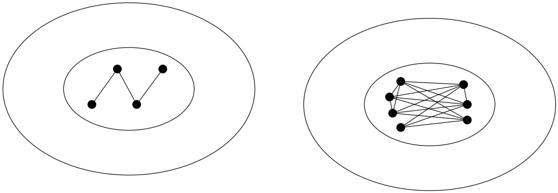
- our P_6-free obstruction has a connected dominating set inducing either a P_4 or a P_4-free graph





Obstructions with arbitrary lists

- our P_6-free obstruction has a connected dominating set inducing either a P_4 or a P_4-free graph

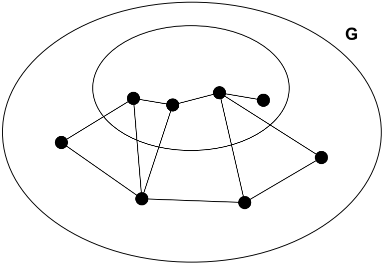


- in the first case we can win by guessing the coloring of the P_4

Obstructions with arbitrary lists

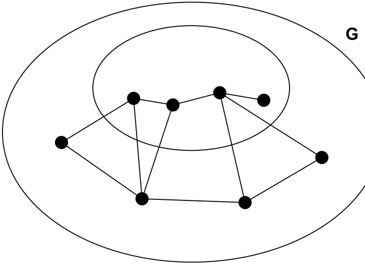
Obstructions with arbitrary lists

- assume there is a dominating P_4



Obstructions with arbitrary lists

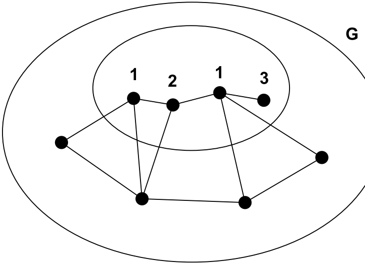
- assume there is a dominating P_4



- We guess its coloring and then reduce the lists of all other vertices. Call the new list system L'.

Obstructions with arbitrary lists

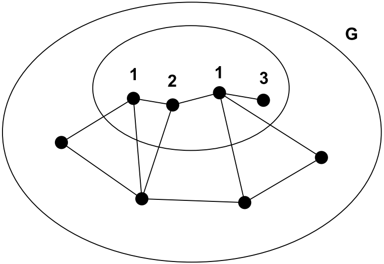
- assume there is a dominating P_4



- We guess its coloring and then reduce the lists of all other vertices. Call the new list system L'.

Obstructions with arbitrary lists

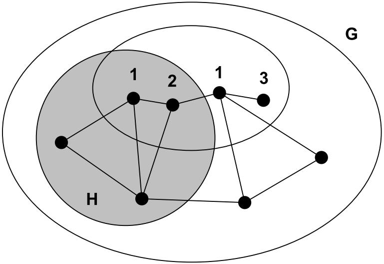
- assume there is a dominating P_4



- We guess its coloring and then reduce the lists of all other vertices. Call the new list system L'.
- in (G, L') each list has at most 2 entries and thus there is a small minimal obstruction H

Obstructions with arbitrary lists

- assume there is a dominating P_4



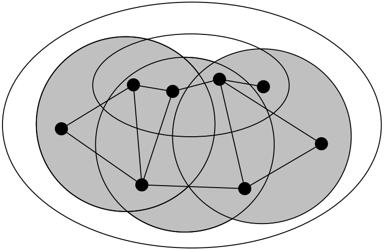
- We guess its coloring and then reduce the lists of all other vertices. Call the new list system L'.
- in (G, L') each list has at most 2 entries and thus there is a small minimal obstruction H

Obstructions with arbitrary lists

- for every coloring of the P_4, we pick a small obstruction H

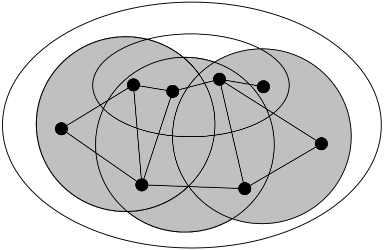
Obstructions with arbitrary lists

- for every coloring of the P_4, we pick a small obstruction H



Obstructions with arbitrary lists

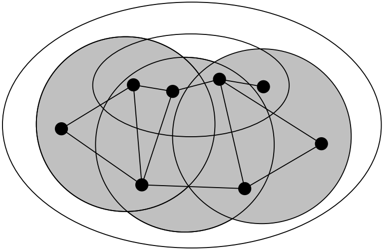
- for every coloring of the P_4, we pick a small obstruction H



- due to the minimality of G, every vertex of G appears in one of the small graphs H

Obstructions with arbitrary lists

- for every coloring of the P_4, we pick a small obstruction H



- due to the minimality of G, every vertex of G appears in one of the small graphs H
- we have constantly many H's, so G has a bounded order

Conclusion

Conclusion

- we want to determine the complexity of k-colorability in H-free graphs

Conclusion

- we want to determine the complexity of k-colorability in H-free graphs
- the results are almost tight, except for 3-colorability...

Conclusion

- we want to determine the complexity of k-colorability in H-free graphs
- the results are almost tight, except for 3-colorability…
- we can describe exactly which cases allow for a finite forbidden induced subgraph characterization

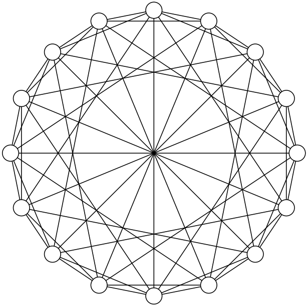
Conclusion

- we want to determine the complexity of k-colorability in H-free graphs
- the results are almost tight, except for 3-colorability. . .
- we can describe exactly which cases allow for a finite forbidden induced subgraph characterization

Thanks!

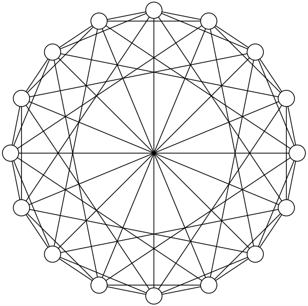
Critical graphs

- There is an infinite family of P_7-free obstructions:



Critical graphs

- There is an infinite family of P_7-free obstructions:



- Easy: infinte familes of claw-free obstructions, and obstructions of large girth
- If H is connected and not a subgraph of P_6, there are infinitely many obstructions in the class of H-free graphs