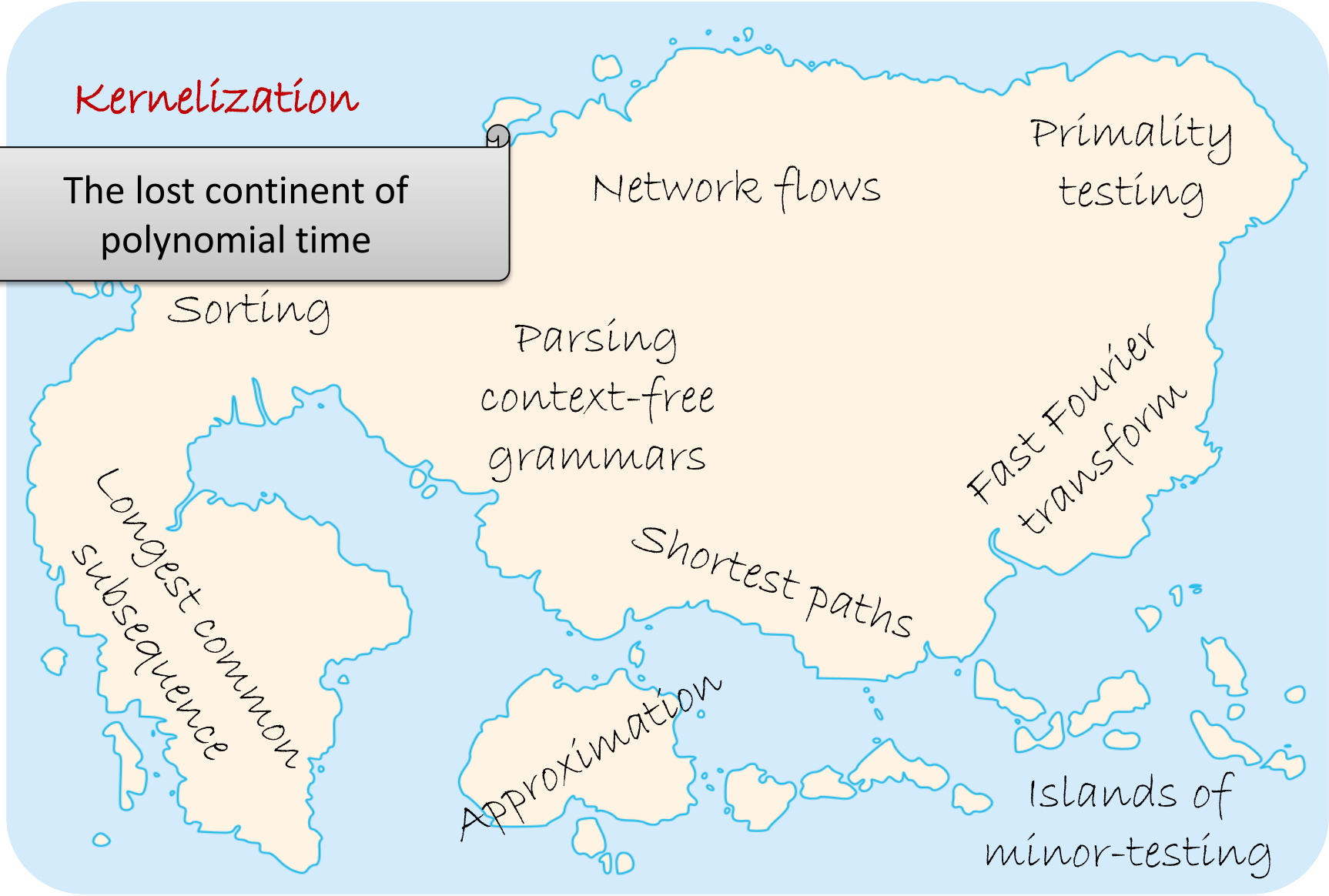
Kernelization: The basics

Bart M. P. Jansen

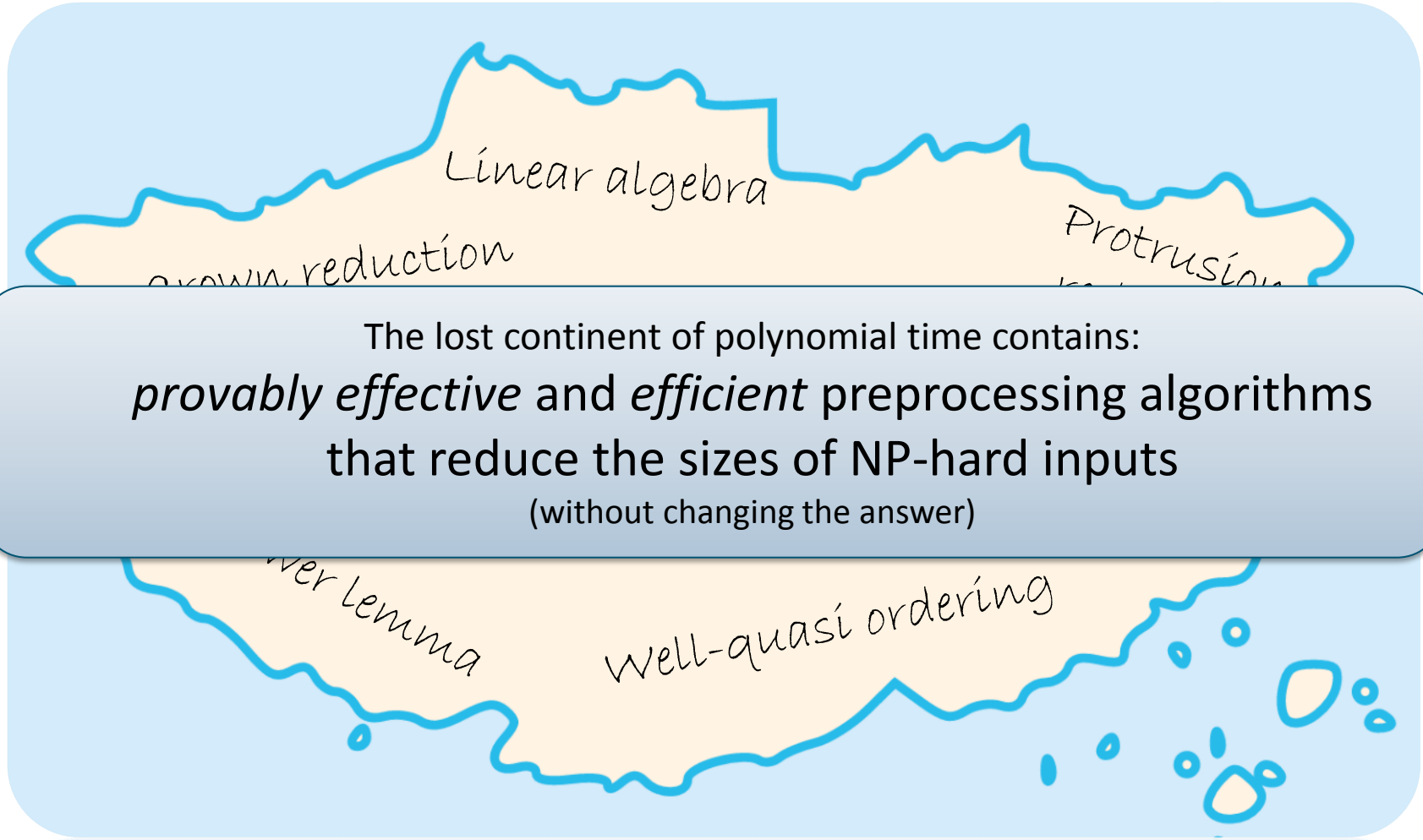
TU/e

Parameterized Complexity Summer School @ Vienna
September 2nd 2017, Vienna, Austria

Map of polynomial-time computation



Map of the lost continent



The lost continent of polynomial time contains:

provably effective and *efficient* preprocessing algorithms that reduce the sizes of NP-hard inputs

(without changing the answer)

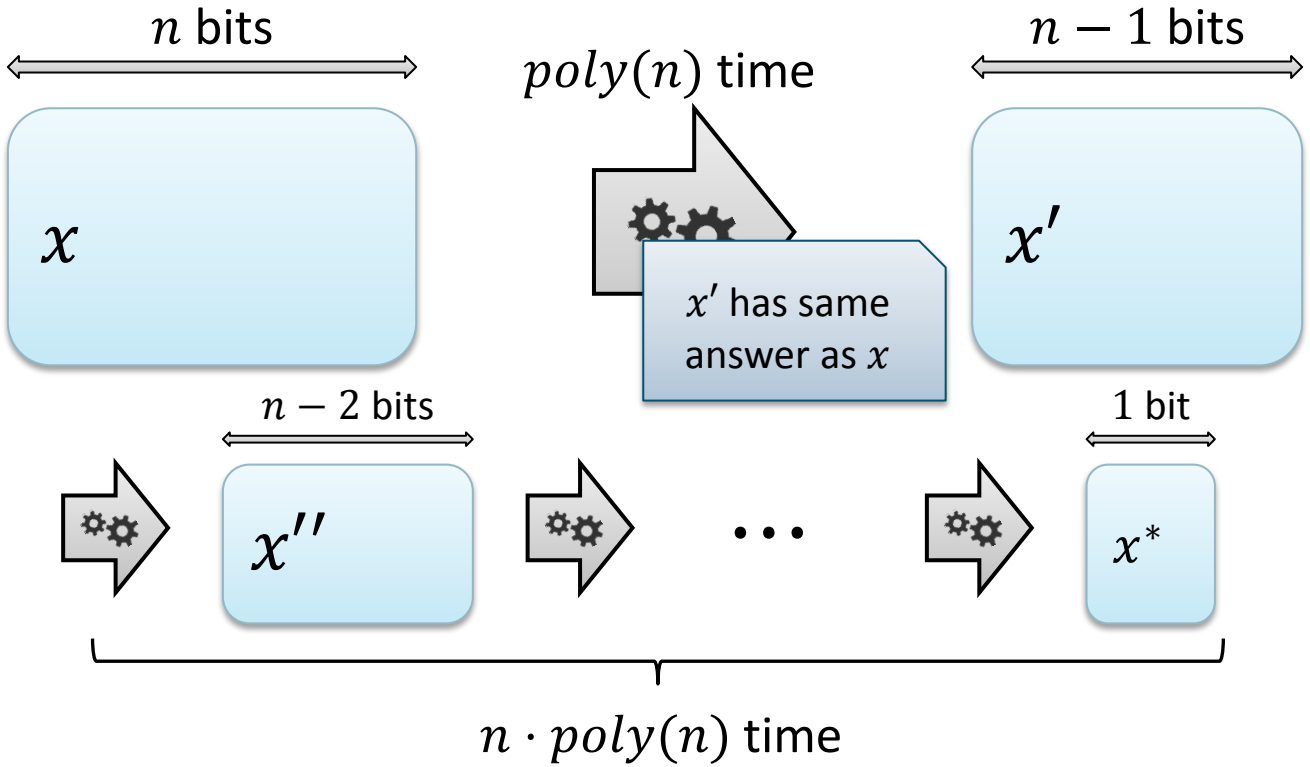
Provable preprocessing

What does *provably effective* and *provably efficient* mean?

Efficient: preprocessing runs in polynomial time

Effective: shrinks the input without changing the answer

How to guarantee an effective preprocessing algorithm?



Provable preprocessing

What does *provably effective* and *provably efficient* mean?

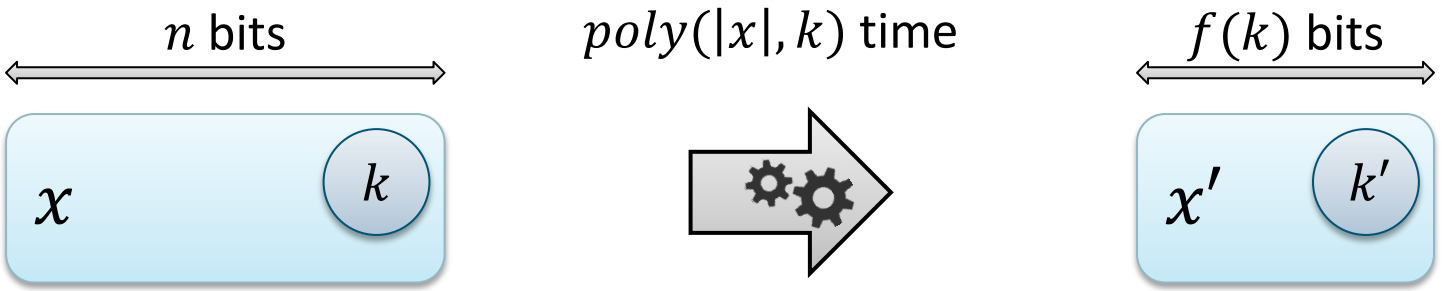
Efficient: preprocessing runs in polynomial time

Effective: shrinks the input without changing the answer

How to guarantee an effective preprocessing algorithm?

If $P \neq NP$, no NP-hard problem has a poly-time preprocessing algorithm that shrinks the input by 1 bit

The viewpoint of *parameterized complexity* helps

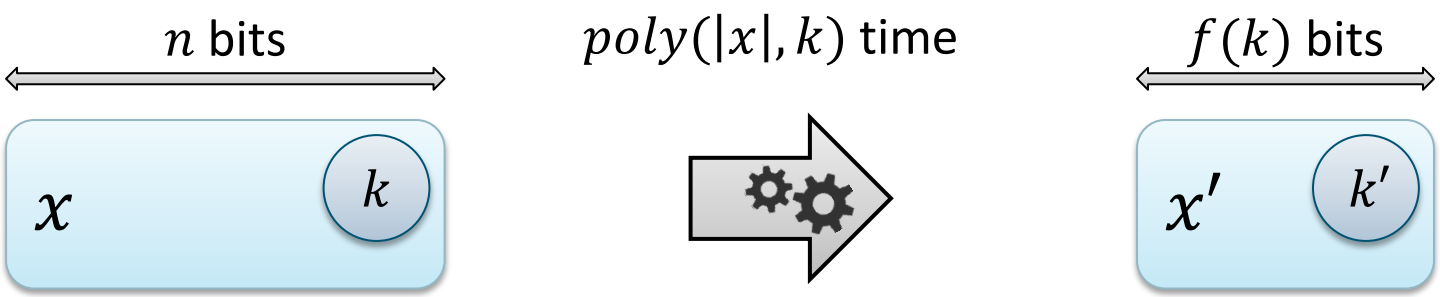


Kernelization: data reduction with a guarantee

A kernelization for a parameterized problem $\mathcal{P}$ is:

- an algorithm that transforms inputs (x, k) into (x', k')
- in $poly(|x|, k)$ time, such that
  - (x, k) has answer YES $\Leftrightarrow$ (x', k') has answer YES, and
  - $|x'| \leq f(k)$ and $k' \leq f(k)$

A kernelization guarantees that instances that are large with respect to their complexity parameter can be shrunk

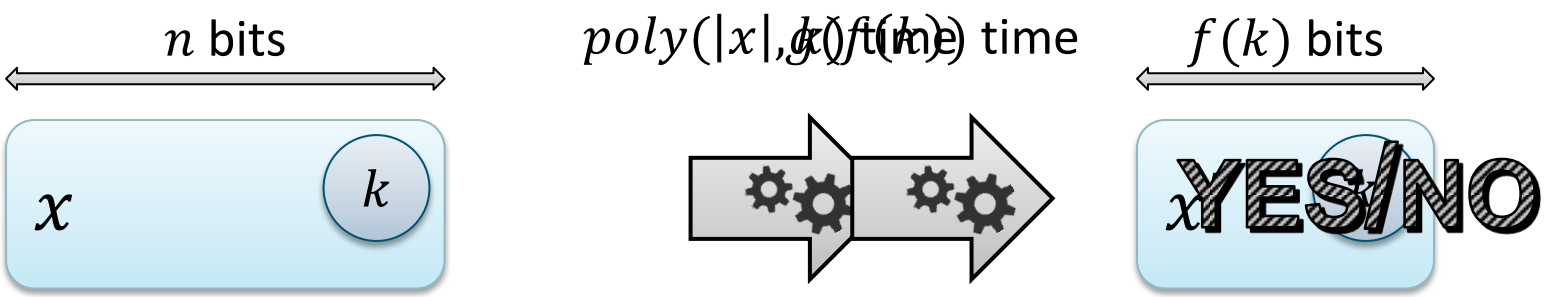


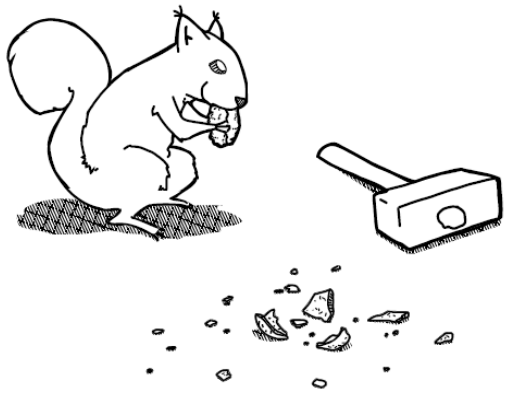
Kernelization & parameterized algorithms

Using any algorithm on the kernel gives an FPT algorithm

For any decidable parameterized problem $\mathcal{P}$:

$\mathcal{P}$ is fixed-parameter tractable $\Leftrightarrow$ $\mathcal{P}$ has a kernel





A KERNEL FOR EDGE CLIQUE COVER

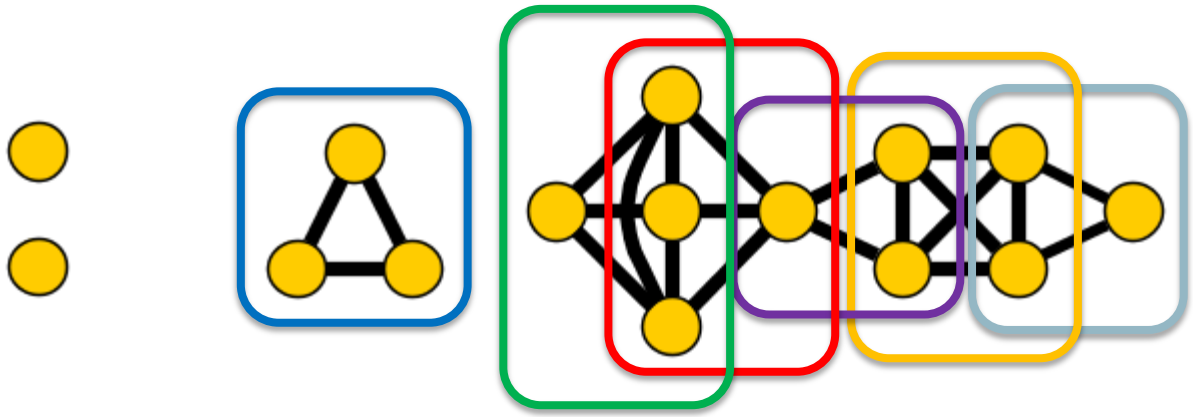
Edge Clique Cover

Input: An undirected graph G and an integer k

Parameter: k

Question: Do there exist k cliques $C_1, \ldots, C_k$ in G, such that for each $\{u, v\} \in E(G)$ there is a clique $C_i \supseteq \{u, v\}$?

Vertices are allowed to belong to more than one clique



YES for $k = 6$

The edge set of G can be covered by k cliques (*is their union*)

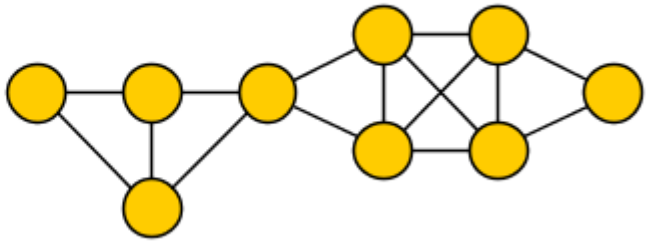
– Notion of cover is different than for Vertex Cover!

Reduction rules for EDGE CLIQUE COVER

(R1) If v is an isolated vertex, then remove v

(R2) If C is a connected component that forms a clique,
then remove C and decrease k by one

(R3) If $N[u] = N[v]$, then remove u (k does not change)

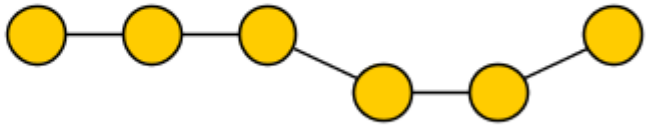


Reduction rules for EDGE CLIQUE COVER

(R1) If v is an isolated vertex, then remove v

(R2) If C is a connected component that forms a clique,
then remove C and decrease k by one

(R3) If $N[u] = N[v]$, then remove u (k does not change)



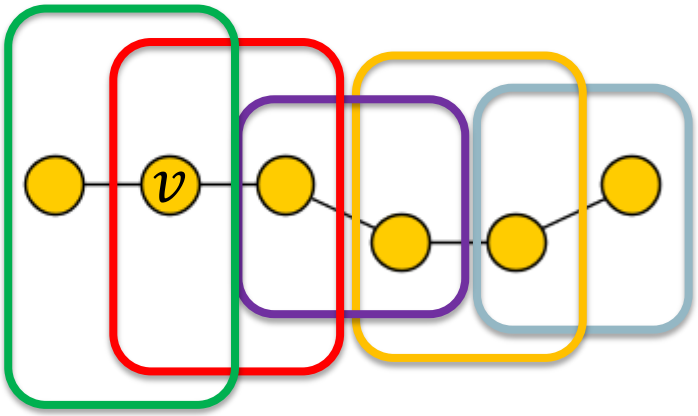
Effectiveness of reduction rules

Lemma. If (G, k) is a YES-instance that cannot be reduced by (R1)-(R3), then:

$$|V(G)| < 2^k$$

Proof. Fix a solution $C_1, \ldots, C_k$ and assign a bitvector to each vertex

$$vec(v) := (v \overset{?}{\in} C_1, v \overset{?}{\in} C_2, \ldots, v \overset{?}{\in} C_k)$$



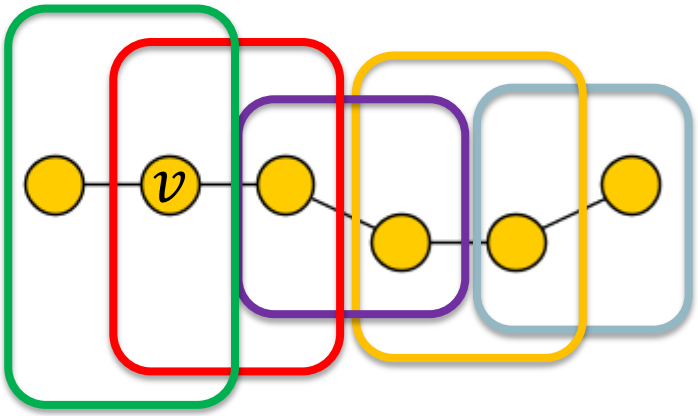
Effectiveness of reduction rules

Lemma. If (G, k) is a YES-instance that cannot be reduced by (R1)-(R3), then:

$$|V(G)| < 2^k$$

Proof. Fix a solution $C_1, \ldots, C_k$ and assign a bitvector to each vertex

$$vec(v) := (1,1,0,0,0)$$



Effectiveness of reduction rules

Lemma. If (G, k) is a YES-instance that cannot be reduced by (R1)-(R3), then:

$$|V(G)| < 2^k$$

Proof. If $vec(v) = 000 \ldots 000$: then v is isolated, (R1) applies

If $vec(v) = vec(u)$: then $N[u] = N[v]$ and (R2/3) applies

So $|V(G)| <$ |{bitvectors of length k}| = 2^k

High-level view of the proof:
If G has a solution, it reveals structure in the graph
If the graph is large wrt. k, then its structured-ness points to an applicable rule

Complete kernelization for Edge Clique Cover

1. Consider input (G, k)
2. Exhaustively apply (R1)-(R3) to obtain (G', k')
3. **if** $|V(G')| \geq 2^{k'}$

   then output "G has no solution of size k"

   else output (G', k') with less than $2^{k'}$ vertices

- This kernelization is essentially the best known

  [Gramm, Guo, Hüffner and Niedermeier, ACM Exper. Alg. 08]

- No kernel of bitsize $2^{o(k)}$ unless P=NP

  [Cygan, Pilipczuk, Pilipczuk, SICOMP'16]

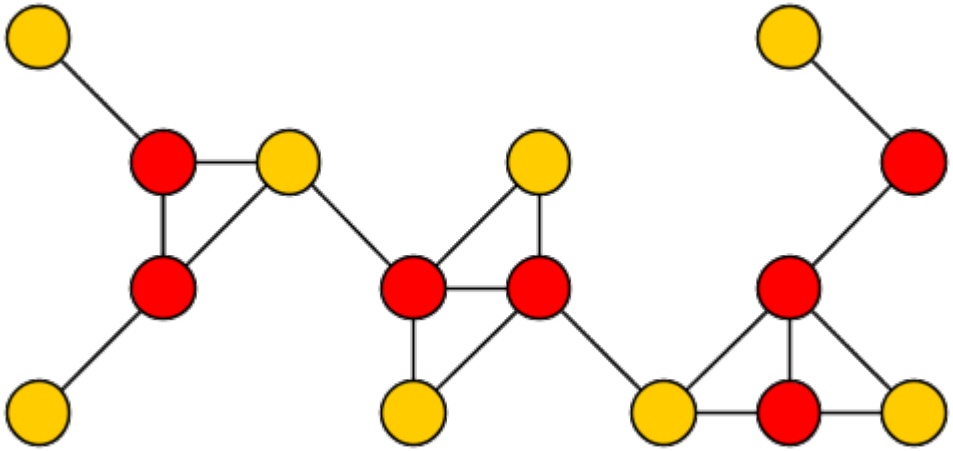
Elementary reduction rules

A KERNEL FOR VERTEX COVER

The VERTEX COVER problem

Input: An undirected graph G and an integer k
Parameter: k
Question: Is there a set S of at most k vertices in G, such that each edge of G has an endpoint in S?

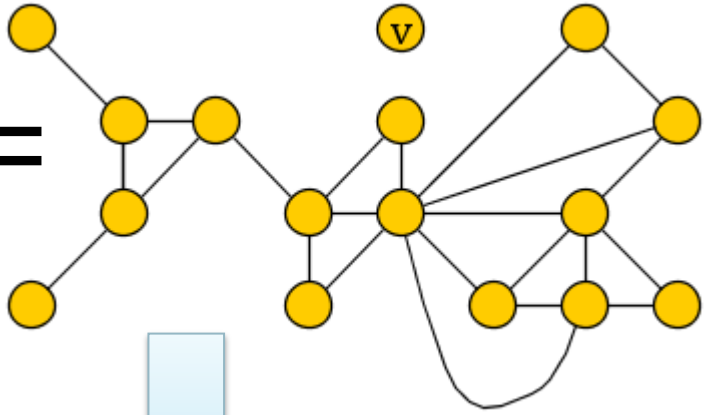
Such a set S is a vertex cover of G



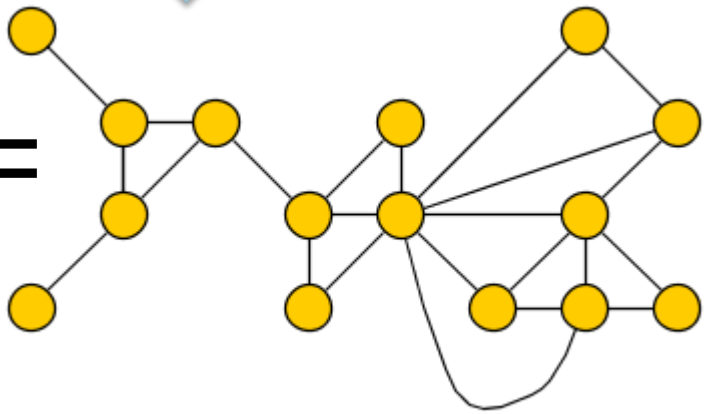
Reduction rules for VERTEX COVER – (R1)

(R1) If there is an isolated vertex v, delete v from G

- Reduce to the instance $(G - v, k)$

$(G =$ 

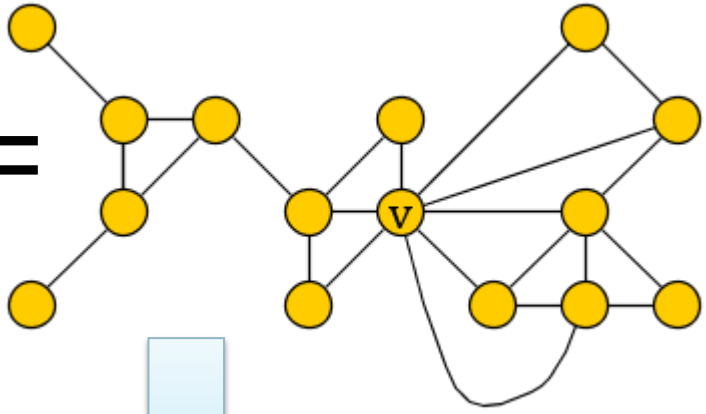
$, k = 7)$

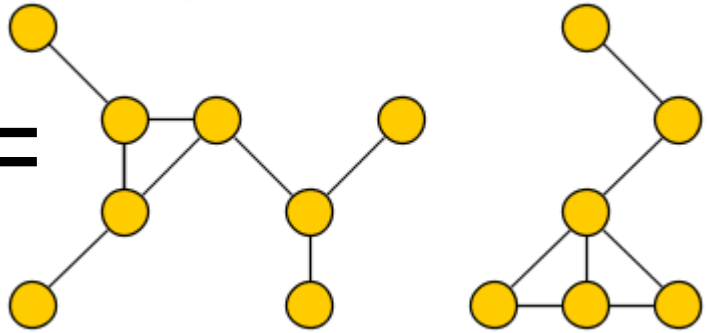
$(G' =$ 
$, k' = 7)$

Reduction rules for VERTEX COVER – (R2)

(R2) If there is a vertex v of degree more than k,
then delete v from G and decrease the parameter by 1

- Reduce to the instance $(G - v, k - 1)$

$(G =$  $, k = 7)$

$(G' =$  $, k' = 6)$

Reduction rules for VERTEX COVER – (R3)

(R3) If the previous rules are not applicable and G has more than $k^2 + k$ vertices or more than k^2 edges, then G has no size-k vertex cover and we output answer NO

Correctness of the cutoff rule

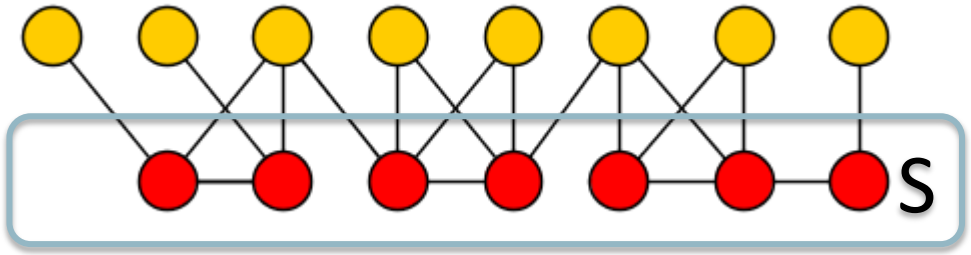
Lemma. If G is exhaustively reduced under (R1)-(R2) and has more than $k^2 + k$ vertices or k^2 edges, then there is no size-$\leq k$ vertex cover

Proof.

- Suppose G has a vertex cover S
- Since (R1) does not apply, every vertex of $G - S$ has at least one edge
- Since (R2) does not apply, every vertex has degree at most k:

$$|E(G)| \leq k \cdot |S|$$
$$|V(G - S)| \leq |E(G)| \leq k \cdot |S|$$

- So $|V(G)| \leq (k + 1) \cdot |S|$
- So if G has a size-k vertex cover, $|V(G)| \leq k^2 + k$ and $|E(G)| \leq k^2$



Preprocessing for VERTEX COVER

(R1)-(R3) can be exhaustively applied in polynomial time

In polynomial time, we reduce (G, k) to (G', k') such that:

- the two instances are equivalent
- $k' \leq k$
- instance (G', k') has at most $k^2 + k$ vertices and k^2 edges

VERTEX COVER parameterized by solution size k has a kernel with $k^2 + k$ vertices and k^2 edges

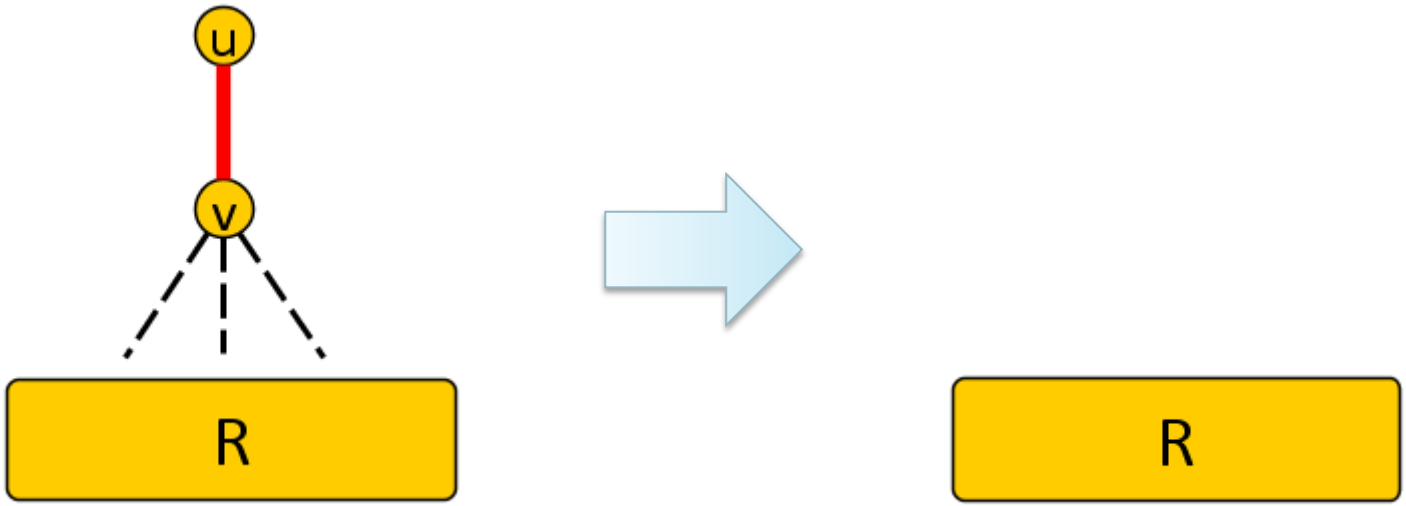
Crown reductions

BETTER KERNEL FOR VERTEX COVER

Motivating examples

If G has a degree-1 vertex u with neighbor v:

- Exists optimal vertex cover using v and not u
- Remove u and v from G to obtain G'
- G has vtx-cover of size $k \Leftrightarrow G'$ has vtx-cover of size $k - 1$

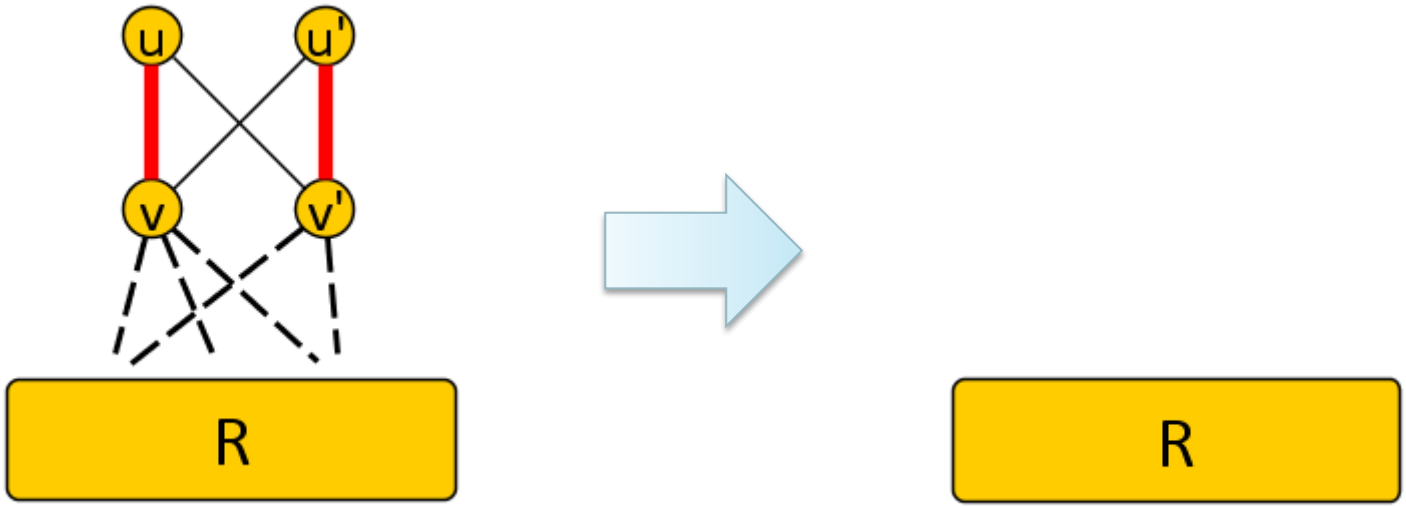


$$k' := k - 1$$

Motivating examples

If G has non-adjacent vertices u, u' with neighborhood $\{v, v'\}$:

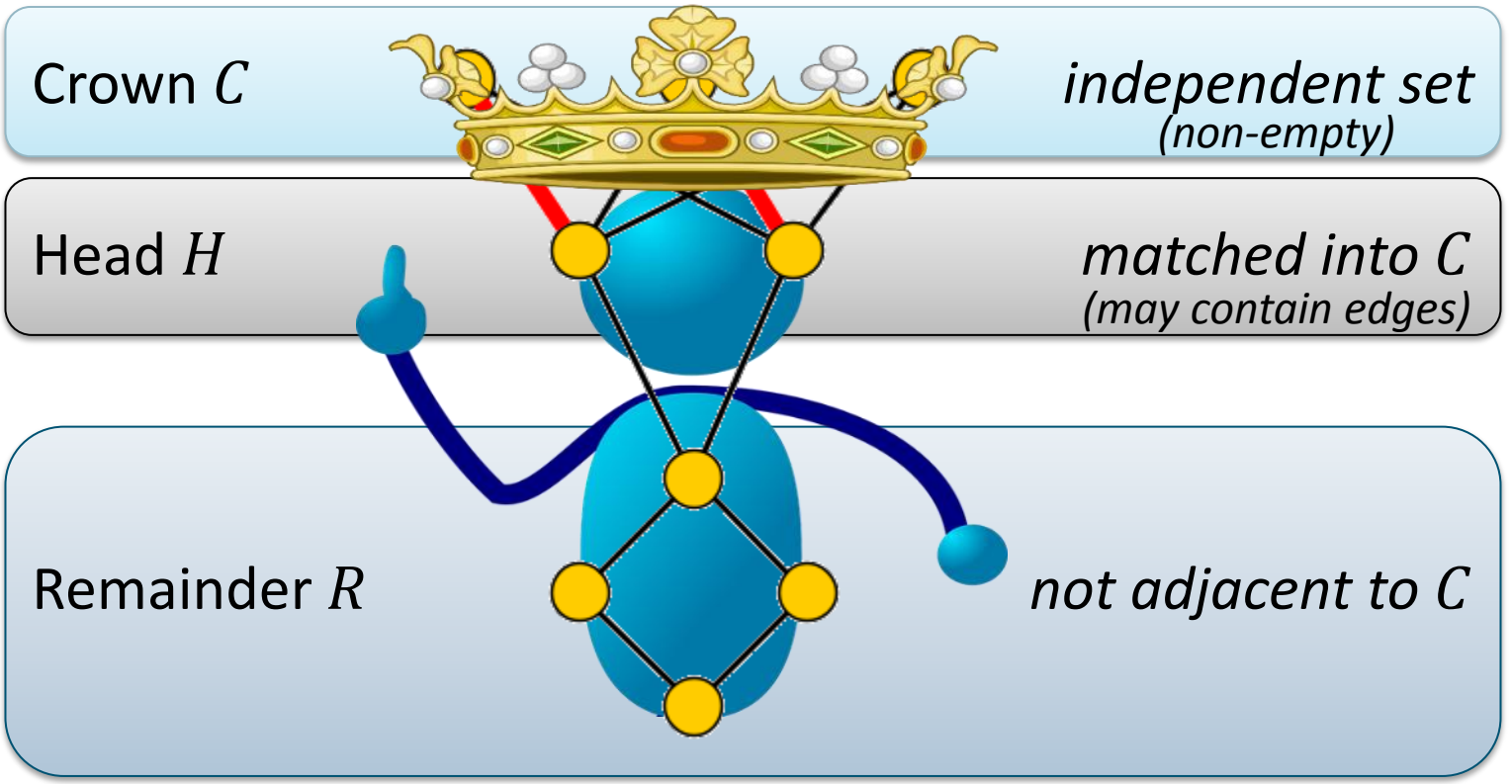
- Exists optimal vertex cover using $\{v, v'\}$ and not $\{u, u'\}$
- Remove $\{u, u', v, v'\}$ from G to obtain G'
- G has vtx-cover of size $k \Leftrightarrow G'$ has vtx-cover of size $k - 2$



$$k' := k - 2$$

Crown decomposition

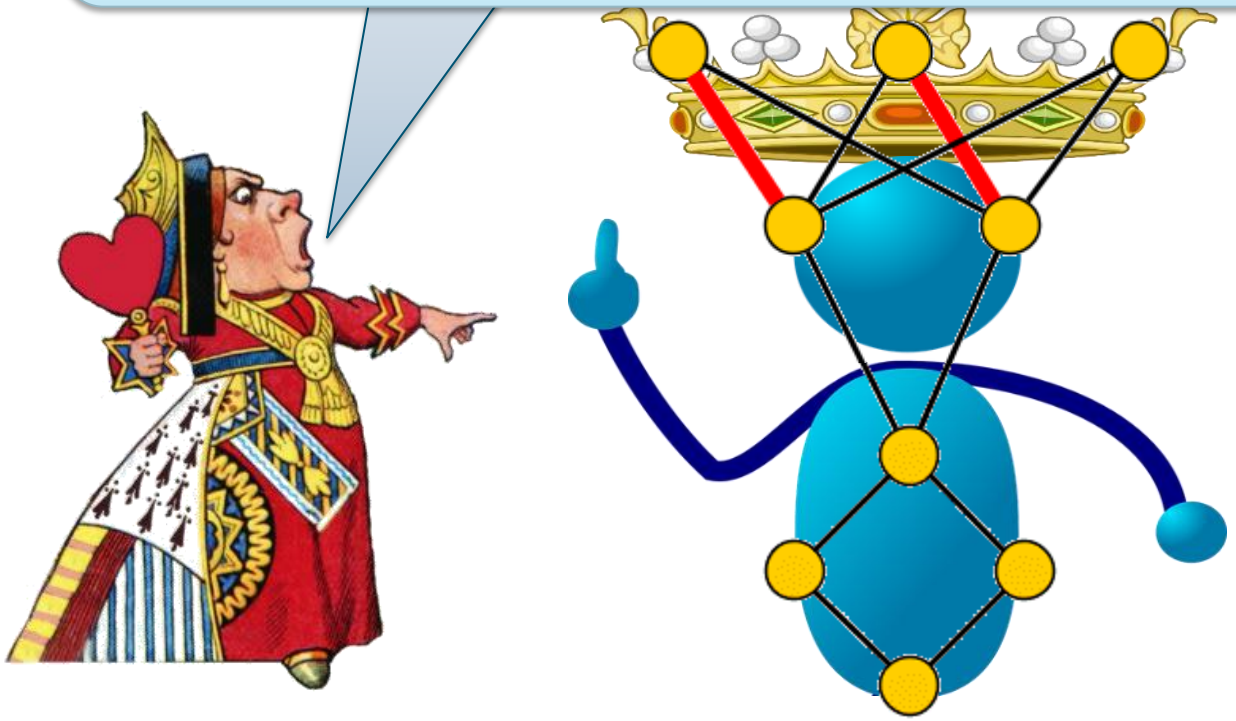
A crown decomposition of graph G is a partition of $V(G)$ into



∃ OPT vertex cover containing all of H and none of C

Crown reduction for VERTEX COVER

G has a vertex cover of size k if and only if
$G - (C \cup H)$ has a vertex cover of size $k - |H|$



Crown-based kernelization for VERTEX COVER

Strategy to kernelize instance (G, k):

1. find a crown decomposition (C, H, R) of $V(G)$
2. remove $C \cup H$ from G
3. decrease k by $|H|$

repeat as long as a crown decomposition can be found

1. How can we find a crown decomposition?

2. What can we guarantee when we can no longer find one?

Crown lemma

If graph G has more than $3k$ vertices, then G has
(a) a matching of size $k+1$, or
(b) a crown decomposition,
and one can be found in polynomial time.

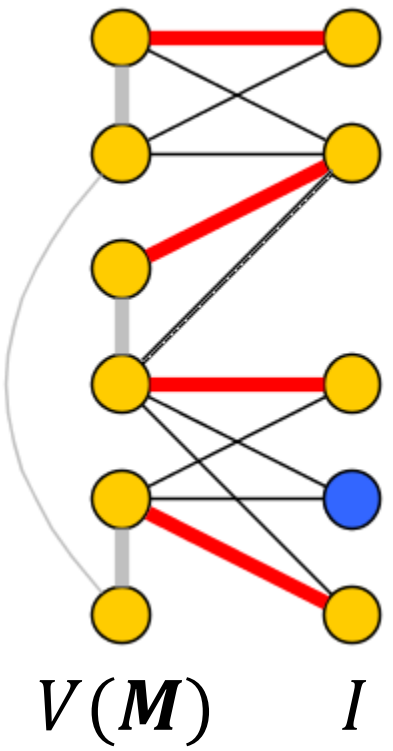
- If we fail to find a crown decomposition of G, then
  - G has at most $3k$ vertices and is kernelized, or
  - G has a matching of size $k+1 \Rightarrow$ no size-k vertex cover

- To get a kernel with $3k$ vertices, suffices to prove the lemma

If graph G has more than $3k$ vertices, then G has
(a) a matching of size $k+1$, or
(b) a crown decomposition,
and one can be found in polynomial time.

- Greedily find maximal matching $\boldsymbol{M}$

1. If $|\boldsymbol{M}| > k$, then (a) holds
2. If $|\boldsymbol{M}| \leq k$, then $|V(\boldsymbol{M})| \leq 2k$
   - Unmatched vertices I are independent
   - Compute maximum matching M' in bipartite graph G' between $V(\boldsymbol{M})$ and I
     i. If $|M'| > k$, then (a) holds
     ii. Else M' leaves a vertex in I unsaturated
     (If M' saturates I, then $V(G) \leq 3k$)



If graph G has more than $3k$ vertices, then G has
(a) a matching of size $k+1$, or
(b) a crown decomposition,
and one can be found in polynomial time.

- Else M' leaves a vertex in I unsaturated

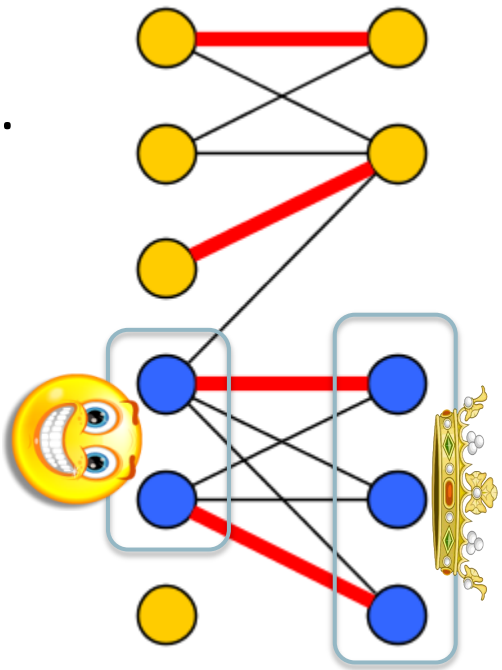
  D := vertices reachable in G' from an unsat. I-vertex by an M'-alternating path

- This gives a crown decomposition:

  Crown: $D \cap I$ [non-empty, independent]

  Head: $D \cap V(M)$ [matched into crown]

  Remainder: $V(G) \setminus D$ [not adjacent to crown]



VERTEX COVER parameterized by solution size k has a kernel with $3k$ vertices and $O(k^2)$ edges

reductions

Linear Kernels in Linear Time, or How to Save k Colors in $O(n^2)$ Steps

WG '04

Benny Chor[1], Mike Fellows[2], and David Juedes[3]

Vertex Cover	Saving k Colors	Max CNF-SAT	Longest Cycle/Path
Disjoint Cycles	Hitting Set	k-Internal spanning tree	Treewidth
Star packing	Triangle packing	Set Packing	P_2-Packing

Open problems for VERTEX COVER kernelization

Current-best kernel has $2k$ vertices [Nemhauser&Trotter'75]

Does VERTEX COVER have a kernel with $(2-\varepsilon)k$ vertices, for any $\varepsilon > 0$?

Current-best runtime for a $O(k)$-vertex kernel is $\Omega(n+m+k^3)$

Does VERTEX COVER have a kernel with $O(k)$ vertices that can be computed in $O(n+m)$ time?

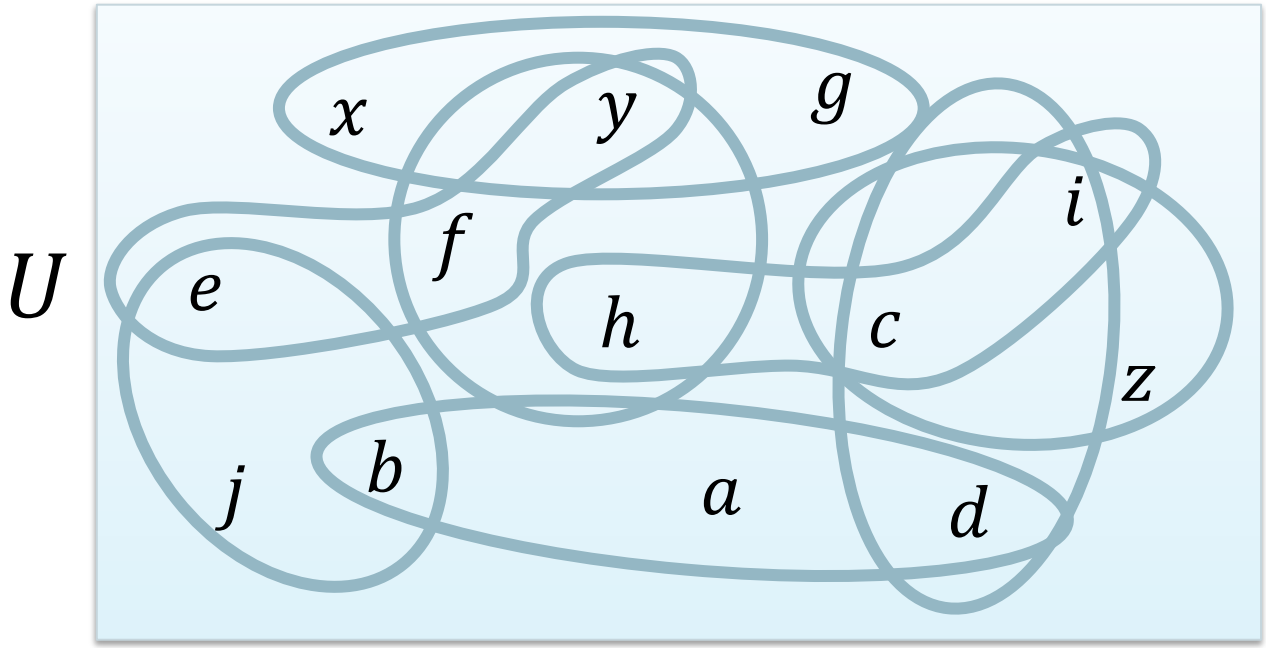
The sunflower lemma

A KERNELIZATION FOR HITTING SET

The d-HITTING SET problem

Input: Set U, integer k, and a collection $\mathcal{F} = F_1, \ldots, F_m$ of subsets of U that have size exactly d

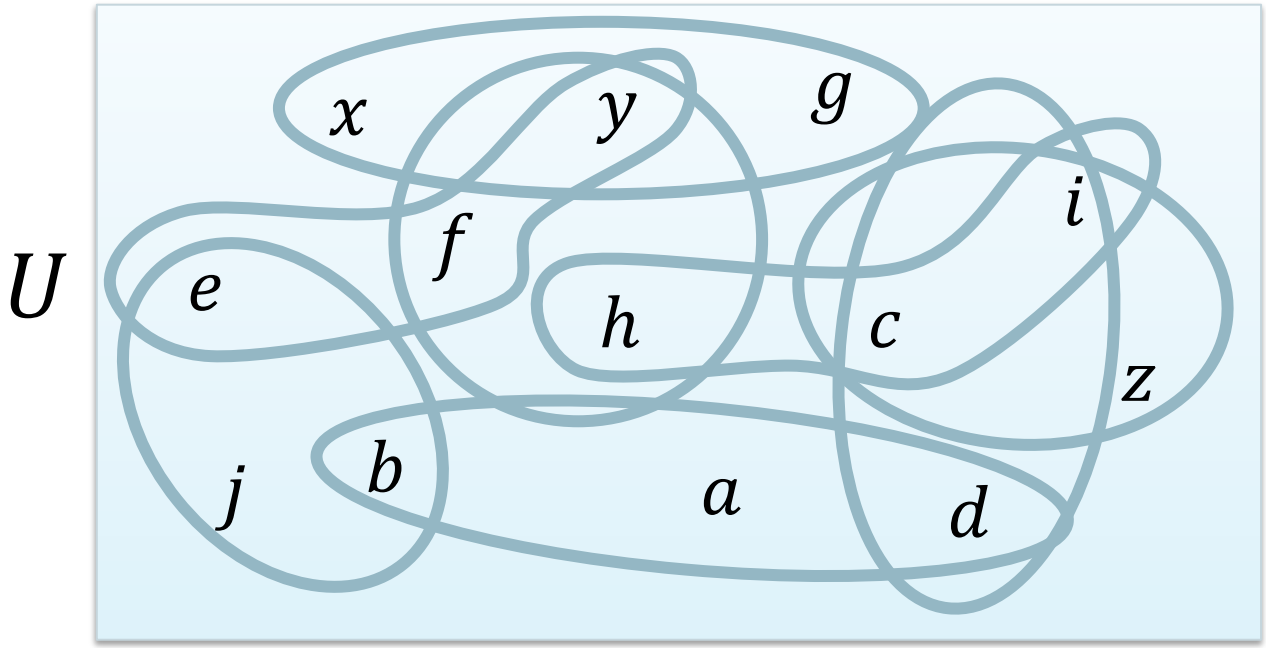
Question: $\exists S \subseteq U$ with $|S| \leq k$ and $S \cap F_i \neq \emptyset$ for all $i \in [m]$?



$S = \{b, c, y\}$ is a hitting set

The d-HITTING SET problem

- The case $d = 2$ corresponds to VERTEX COVER
- Can express DOMINATING SET in graphs of max degree $d - 1$
  - W[2]-complete without the restriction to size-d sets



$S = \{b, c, y\}$ is a hitting set

Sunflowers

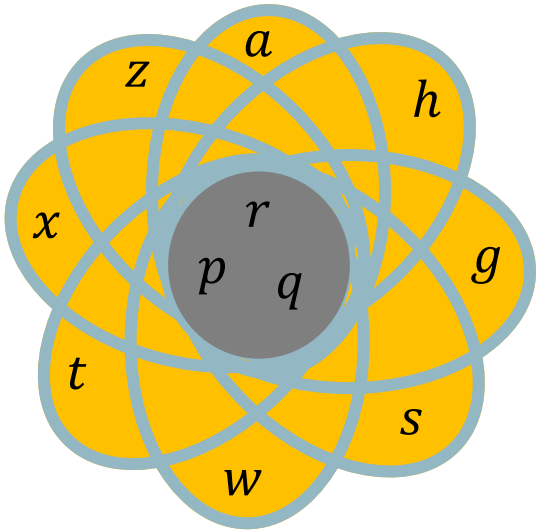
A sunflower with core C is a collection of sets $F_1, \dots, F_\ell$ with:

$F_i \cap F_j = C$ for all distinct $i, j \in [m]$

$F_i \setminus C \neq \emptyset$ for all $i \in [m]$

The sets $F_i \setminus C$ are the petals, they are pairwise disjoint

$$C = \{p, q, r\}$$



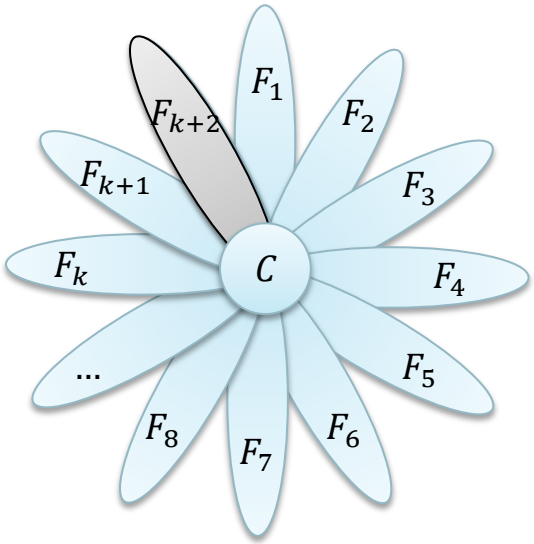
C is allowed to be empty

Sunflower reduction rule for d-Hitting Set

If $\mathcal{F}$ has a sunflower $F_1, \ldots, F_{k+2}$ with core C and $k+2$ petals:

Reduce to $\mathcal{F}' := \mathcal{F} \setminus \{F_{k+2}\}$

- If S is a hitting set of size at most k for $\mathcal{F}'$:
  - The petals $F_1 \setminus C, \ldots, F_{k+1} \setminus C$ are disjoint
  - Since S hits $F_1, \ldots, F_{k+1}$ with k vertices: $S \cap C \neq \emptyset$
  - Hence $S \cap F_{k+2} \neq \emptyset$

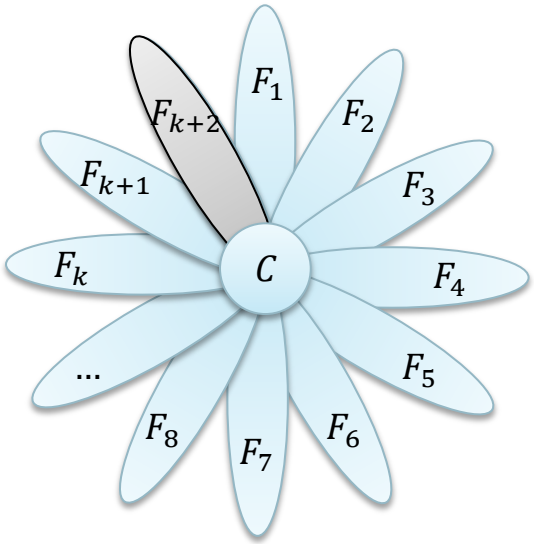


Sunflower reduction rule for d-HITTING SET

If $\mathcal{F}$ has a sunflower $F_1, \ldots, F_{k+2}$ with core C and $k+2$ petals:

Reduce to $\mathcal{F}' := \mathcal{F} \setminus \{F_{k+2}\}$

- If S is a hitting set of size at most k for $\mathcal{F}'$, then it also hits $\mathcal{F}$
- If S is a hitting set of size at most k for $\mathcal{F}$, then it also hits $\mathcal{F}'$



Kernelization strategy for d-HITTING SET

While there is a sunflower $F_1, \ldots, F_{k+2}$ with $k + 2$ petals:

- Remove F_{k+2} from the collection of sets that must be hit

1. How can we find a sunflower with $k + 2$ petals?

2. What can we guarantee when we can no longer find one?

Finding sunflowers

Let $\mathcal{F}$ be a collection of sets of size exactly d

Sunflower lemma. If $|\mathcal{F}| > d!\,(k+1)^d$,
then $\mathcal{F}$ contains a sunflower with $k+2$ petals.
It can be found in polynomial time.

If the reduction rule cannot be applied, we have:

$$|\mathcal{F}| \leq d!\,(k+1)^d = O\left(k^d\right)$$

Threshold is independent of the universe size

Kernel for d-Hitting Set

1. Sunflower rule ensures $|\mathcal{F}| \leq O(k^d)$
2. Remove elements from U that do not occur in any set in $\mathcal{F}$
   - Ensures $|U| \leq d \cdot |\mathcal{F}| = O(k^d)$

d-Hitting Set has a kernel with $O(k^d)$ sets and elements

- Extends to a kernel for setting where each set in $\mathcal{F}$ has size at most d instead of exactly d

Open problem for HITTING SET kernelization

Current-best kernel has $O(k^{d-1})$ elements and $O(k^d)$ sets
[Abu-Khzam, JCSS'10]

Does d-HITTING SET have a kernel with $f(d)k$ elements?

WRAP-UP

Mindsets when developing kernels

Reduction-rule first

- Try to come up with provably safe reduction rules
- Analyze what a large irreducible instance looks like

Structure first

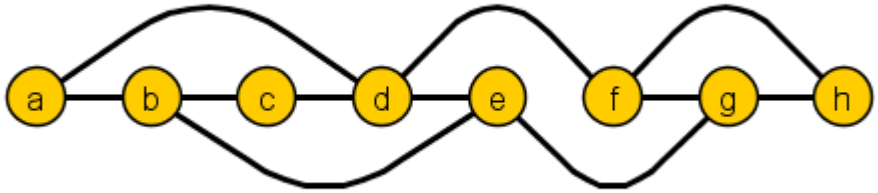
- Investigate what a large instance looks like to which the answer is not obviously YES or obviously NO
- Develop reduction rules to attack the `large parts'

Extremal structure of minimal obstructions

- For graph problems that become easier for subgraphs
- How large can a graph G be in terms of k, when (G, k) is NO but all subgraphs $(G' \subseteq G, k)$ yield YES?

One more open problem

Consider a linear ordering of the vertex set of a graph G



The imbalance $\sigma(v)$ of a vertex v is the absolute value of:

$$|N(v) \cap \{\text{vtces before } v\}| - |N(v) \cap \{\text{vertices after } v\}|$$

The imbalance of the linear ordering is $\sum_{v \in V(G)} \sigma(v)$

IMBALANCE asks: does G have an ordering of imbalance $\leq k$?

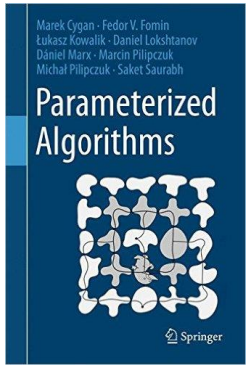
Does IMBALANCE have a polynomial kernel?

http://fptschool.mimuw.edu.pl/opl.pdf

Exercises

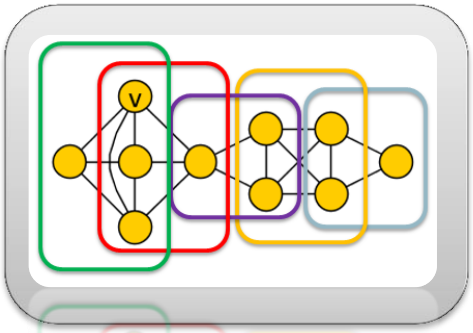
- Build an elementary kernel for CLUSTER EDITING [Ex. 2.5]
- Build an exponential kernel for CONNECTED VERTEX COVER [Ex. 2.14]
- Build a kernel for DUAL COLORING using crowns [Ex. 2.22]
- Build a kernel for d-SET PACKING using sunflowers [Ex. 2.29]
- Prove the sunflower lemma using induction on d
  Hint: Use a greedy packing of disjoint sets [Thm. 2.25]

References to *Parameterized Algorithms* by Cygan et al.
http://link.springer.com/book/10.1007%2F978-3-319-21275-3

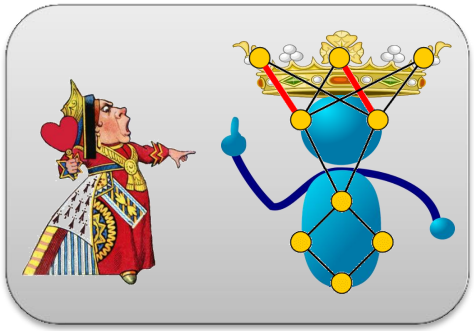


Summary

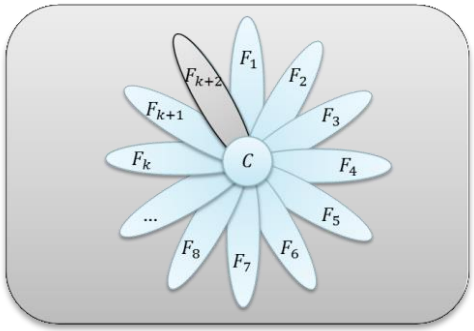
Several kernelizations using interesting combinatorial insights:



EDGE CLIQUE COVER



VERTEX COVER



d-HITTING SET

Kernelization enables a rigorous scientific study of preprocessing
&
forms a route to fixed-parameter tractable algorithms

THANK YOU!