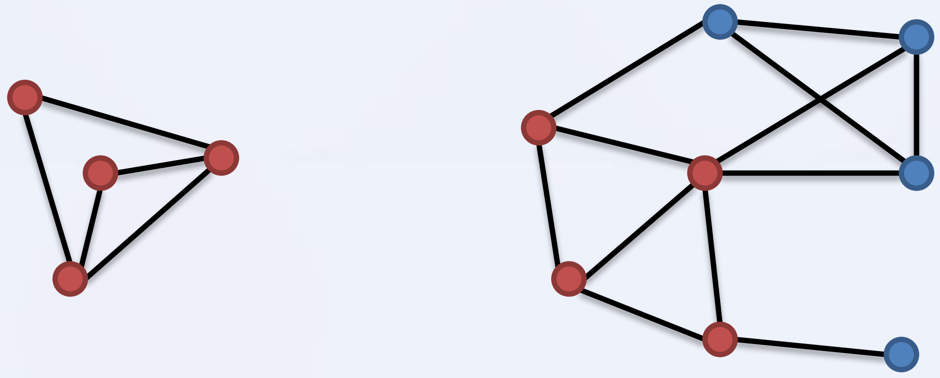
Subexponential Time Algorithms for Embedding H-Minor Free Graphs

Hans Bodlaender, Jesper Nederlof, Tom van der Zanden

Subgraph Isomorphism

- Given a *pattern* graph P and *host* graph G
- Decide whether G has subgraph isomorphic to P



- Let $n = |V(G)|$ and $k = |V(P)|$

Context

- **Exponential Time Hypothesis**

  n-variabele 3-SAT can not be solved in time $2^{o(n)}$

- **“Square Root Phenomenon”**

  Problems easier on planar graphs, e.g.:

  - 3-Coloring/Hamiltonian Cycle...: $2^{O(n)} \rightarrow 2^{O(\sqrt{n})}$
  - Independent Set/...: $n^{O(k)} \rightarrow n^{O(\sqrt{k})}$

- Tight under ETH

Subgraph Isomorphism?

- General: No $2^{o(n \log n)}$ algorithm [Fomin et al. ‘15]
- Planar graphs: $2^{O(k)}n$ algorithm [Dorn ‘09]

Does square root phenomenon hold?

Our results

- **Algorithm**: Subgraph Isomorphism can be solved in $2^{O(n\,/\,\log n)}$ time on planar graphs

Our results

- **Algorithm**: Subgraph Isomorphism can be solved in $2^{O(n / \log n)}$ time on planar graphs
- **Lower bound**: there is no $2^{o(n / \log n)}$ time algorithm unless ETH fails

- (Slightly more general): $2^{O(k^{\epsilon}\, tw(G) + k / \log k)} n^{O(1)}$ if P is H-Minor Free for any $\epsilon > 0$ and fixed graph H
- (Induced) Subgraph, (Induced) Minor, Topological Minor,…

Lower Bound

- Reduce from 3-SAT with $m = O(n)$ clauses
- Variables $x_1, \ldots x_n$; Clauses $c_{n+1,\ldots,}c_{n+m+1}$

- No $2^{o(n)}$ algorithm by Sparsification Lemma

Lower Bound

- Reduce from 3-SAT with $m = O(n)$ clauses
- Variables $x_1, \ldots x_n$; Clauses $c_{n+1,\ldots,}c_{n+m+1}$
- Build bitstrings for variables/clauses

00001 11110 01111 10000

Index of C/V in binary

- Length $O(\log n)$

Lower Bound

- Reduce from 3-SAT with $m = O(n)$ clauses
- Variables $x_1, \ldots x_n$; Clauses $c_{n+1,\ldots,}c_{n+m+1}$
- Build bitstrings for variables/clauses

00001 11110 01111 10000

Bitwise complement

- Length $O(\log n)$

Lower Bound

- Reduce from 3-SAT with $m = O(n)$ clauses
- Variables $x_1, \ldots x_n$; Clauses $c_{n+1,\ldots,}c_{n+m+1}$
- Build bitstrings for variables/clauses

00001 11110 01111 10000

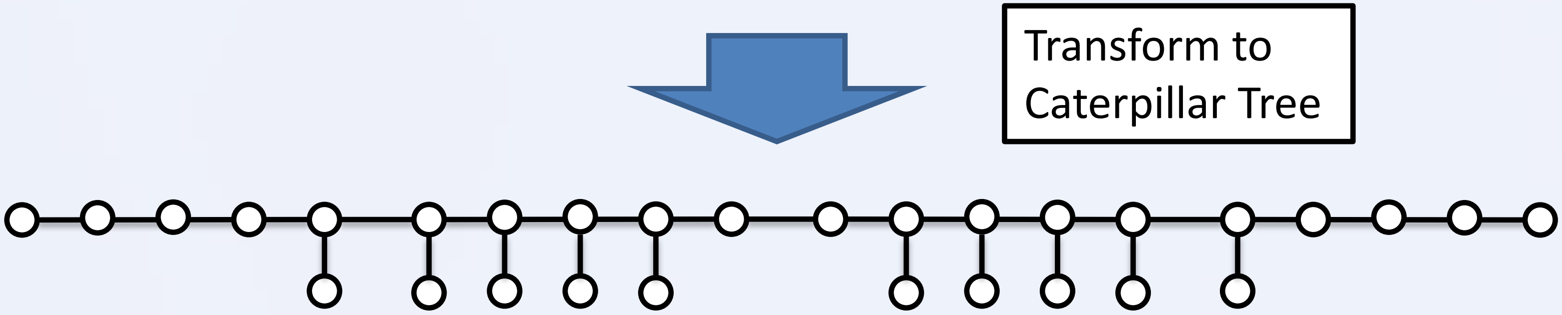
Make Palindromic

- Length $O(\log n)$

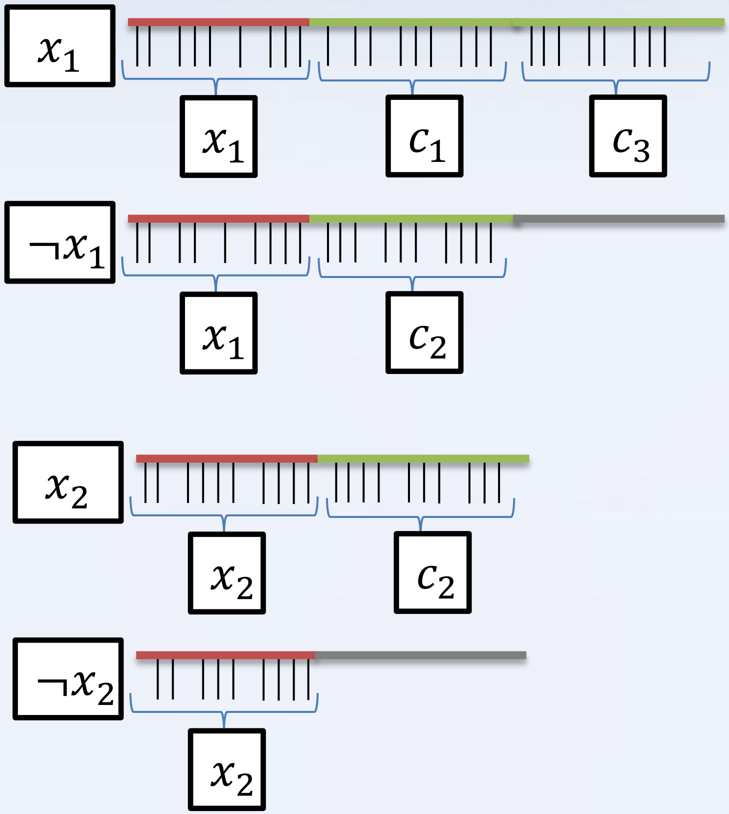
Lower Bound

- Reduce from 3-SAT with $m = O(n)$ clauses
- Variables $x_1, \ldots x_n$; Clauses $c_{n+1,\ldots,}c_{n+m+1}$
- Build bitstrings for variables/clauses

00001 11110 01111 10000

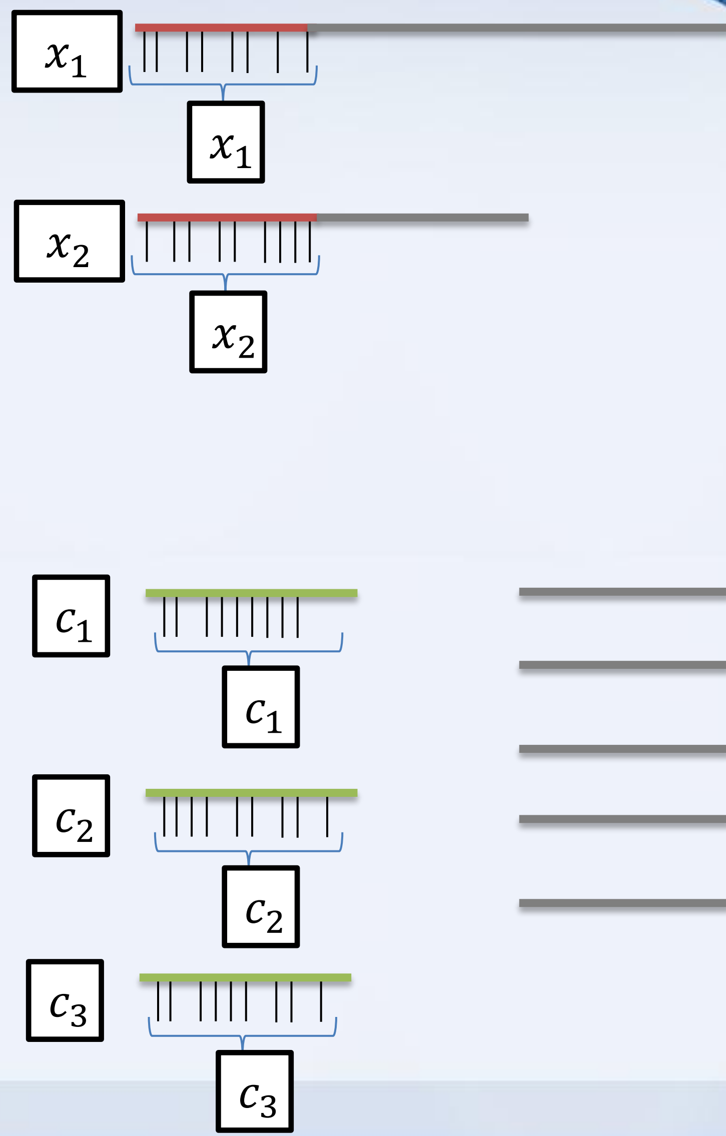


Host graph G

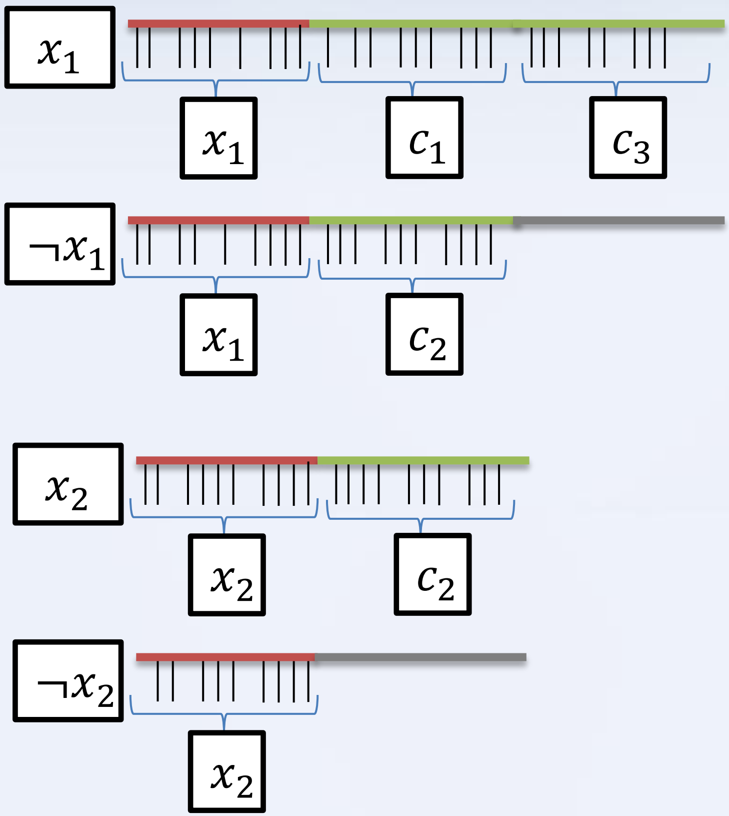


$(x_1 \lor x_3 \lor x_4) \land (\neg x_1 \lor x_2 \lor x_5) \land (x_1 \lor \cdots$

Guest graph P

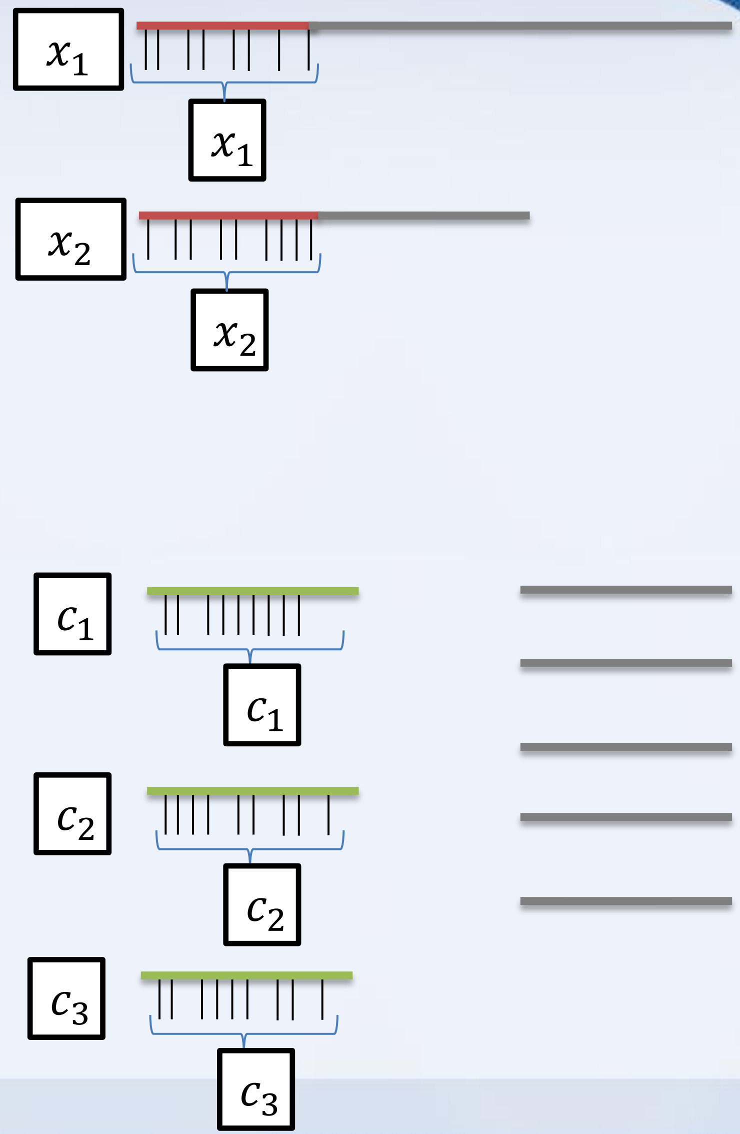


Host graph G



$(x_1 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_5) \wedge (x_1 \vee \cdots$

Guest graph P



Lower Bound

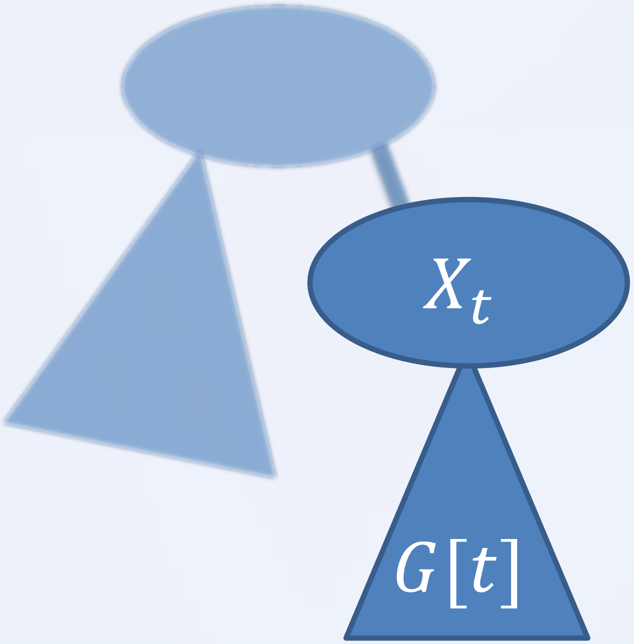
- Collection of caterpillar trees
- Size of instance: $O(n \log n)$
  $\rightarrow 2^{\Omega(n / \log n)}$ lower bound
- Can add 'universal' vertex to make P a tree and G connected, pathwidth 2

Algorithm

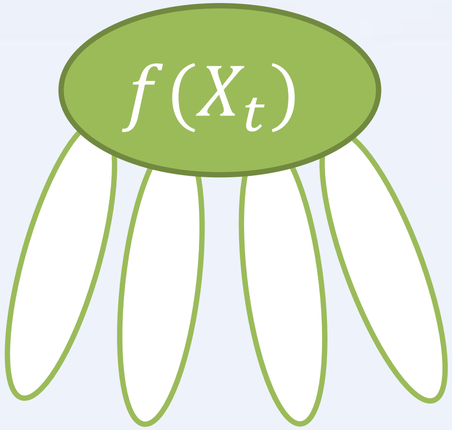
- Dynamic Programming on Tree Decomposition of host graph G

Builds upon [Hajiaghayi 2007]

Host graph $\boldsymbol{G}$



Guest graph P

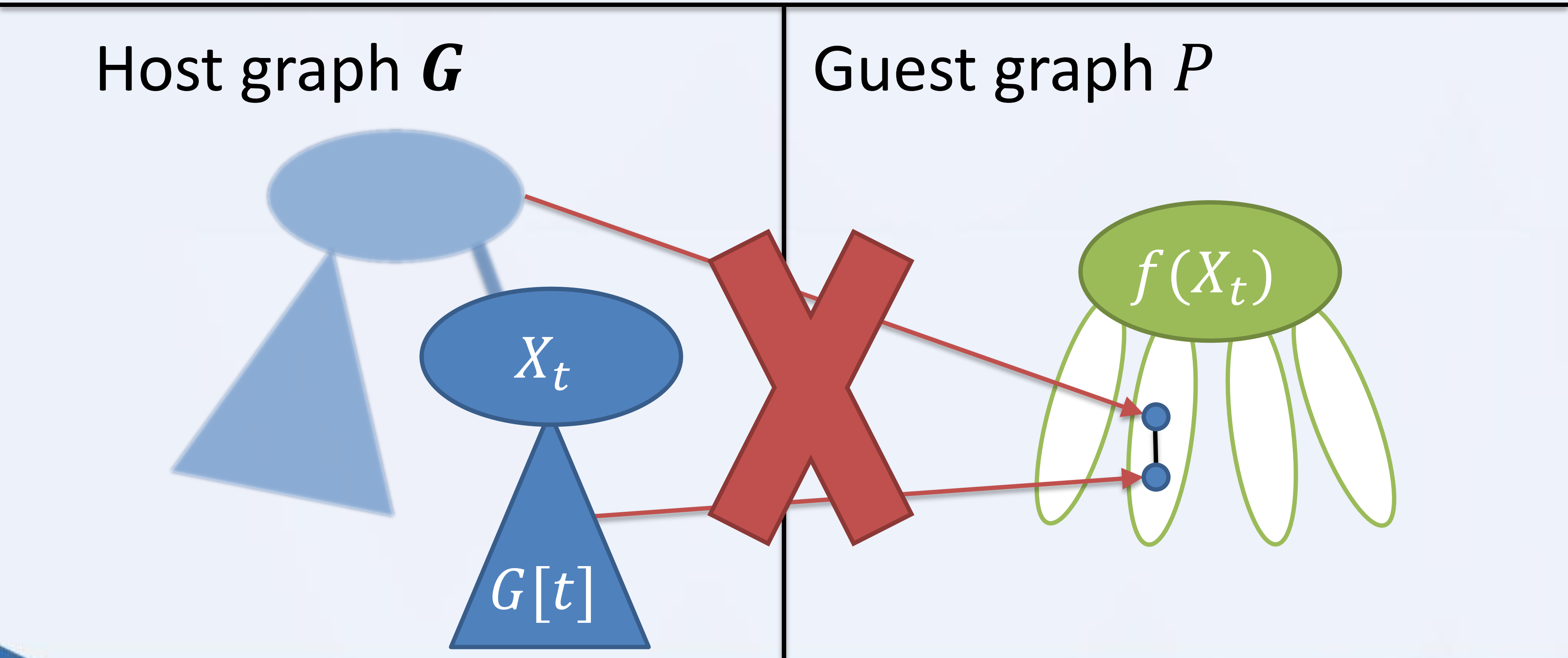


- Partial solution: $f: G[t] \rightarrow V(P) \cup \{\square\}$

Algorithm

- Dynamic Programming on Tree Decomposition of host graph G

Builds upon [Hajiaghayi 2007]



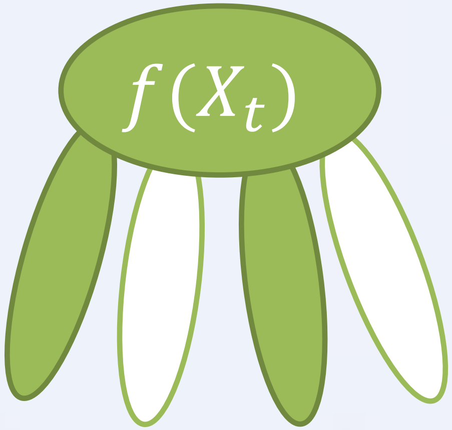
- Partial solution: $f: G[t] \rightarrow V(P) \cup \{\square\}$

Partial Solution

- Keeping track of $f(X_t)$: $n^{O(\sqrt{n})}$ options
- Subset of connected components $V(P) \setminus f(X_t)$

$$2^{\#\text{components}} = 2^n \text{ worst case!!}$$

- Only trouble for **small** components



Big and Small

- A connected component S of $V(P) - f(X_t)$ is
  - *Small* if $|V(S)| \leq c \log n$
  - *Big* if $|V(S)| > c \log n$

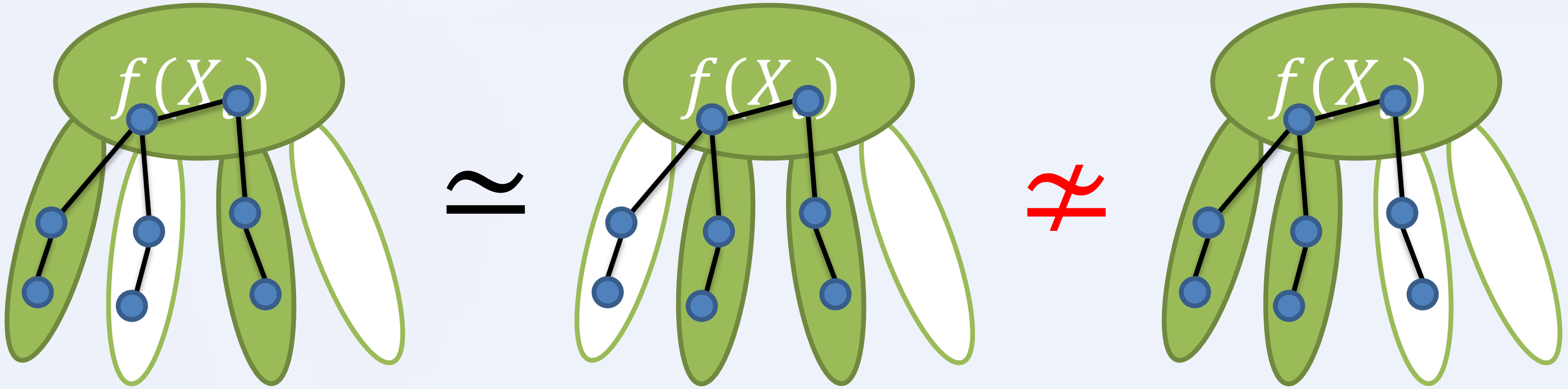
- At most $\frac{\mathrm{n}}{\mathrm{c} \log \mathrm{n}}$ big components: $2^{O(n/\log n)}$

Small Components

- How many cases for small components?
- There exists C, such that number of non-isomorphic r-vertex planar graphs is at most C^r
- For small enough c, at most $C^{c \log n} = n^{\epsilon}$
- Specify how many of each: $n^{(n^{\epsilon})}$

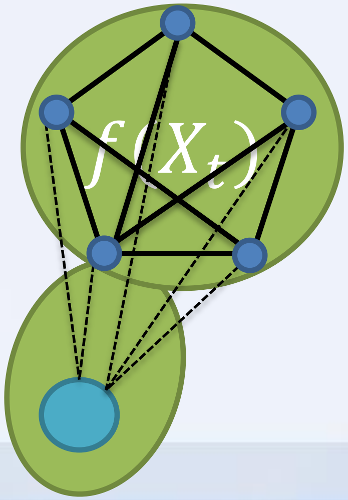
Component Isomorphism

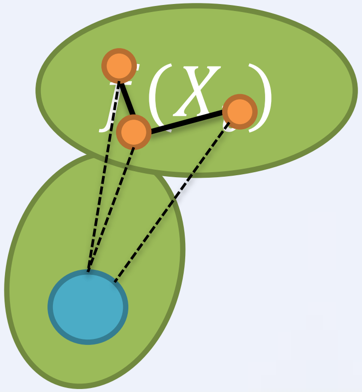
- Partial Solutions are **equivalent** if they have the same number of components from each **component isomorphism class**



Neighborhoods

- Gajarský et al. (2013) show that
  - Few connected components have a large neighborhood;
  - Remaining components have few **distinct** small neighborhoods.





Isomorphism Classes

- At most $tw(G)k^{\epsilon}$ isomorphism classes of small cc's
- Count how many of each: at most $k^{tw(G)k^{\epsilon}}$
- $2^{O(n/\log n)}$ for big components dominates running time
- Tradeoff between big and small

Conclusions

- $2^{\Theta(n/\log n)}$ algorithm and lower bound for Subgraph Isomorphism on Planar Graphs
- Also possible: induced subgraph, (induced) minor
- Can we do $2^{O(k/\log k)}n^{O(1)}$? Lower bound?
  - Partial answer by Fomin et al. [FOCS 2016]
- Maybe possible: shallow minor, topological minor, immersion,…
- Other problems where isomorphism helps?