Linear kernels for edge deletion problems to immersion-closed graph classes

Jean-Florent Raymond

LIRMM and University of Warsaw

Journées du GT COA 2016

Joint work with Archontia Giannopoulou (TU Berlin), Michał Pilipczuk (University of Warsaw), Dimitrios M. Thilikos (LIRMM–CNRS), and Marcin Wrochna (University of Warsaw).

What is a deletion problem?

Question

Can I delete a few vertices/edges in my graph and get a nice property?

What is a deletion problem?

Question

Can I delete a few vertices/edges in my graph and get a nice property?

The properties we consider: to be $\mathcal{F}$-minor-free or $\mathcal{F}$-immersion-free

What is a deletion problem?

Question

Can I delete a few vertices/edges in my graph and get a nice property?

The properties we consider: to be $\mathcal{F}$-minor-free or $\mathcal{F}$-immersion-free

$\mathcal{F}$-Minor-Deletion

Input: a graph G and an integer k;
Question: can I remove in G k vertices to get an $\mathcal{F}$-minor-free graph?

What is a deletion problem?

Question

Can I delete a few vertices/edges in my graph and get a nice property?

The properties we consider: to be $\mathcal{F}$-minor-free or $\mathcal{F}$-immersion-free

$\mathcal{F}$-Minor-Deletion

Input: a graph G and an integer k;
Question: can I remove in G k vertices to get an $\mathcal{F}$-minor-free graph?

$\mathcal{F}$-Immersion-Deletion

Input: a graph G and an integer k;
Question: can I remove in G k edges to get an $\mathcal{F}$-immersion-free graph?

What is a deletion problem?

Question

Can I delete a few vertices/edges in my graph and get a nice property?

The properties we consider: to be $\mathcal{F}$-minor-free or $\mathcal{F}$-immersion-free

$\mathcal{F}$-Minor-Deletion

Input: a graph G and an integer k;
Question: can I remove in G k vertices to get an $\mathcal{F}$-minor-free graph?

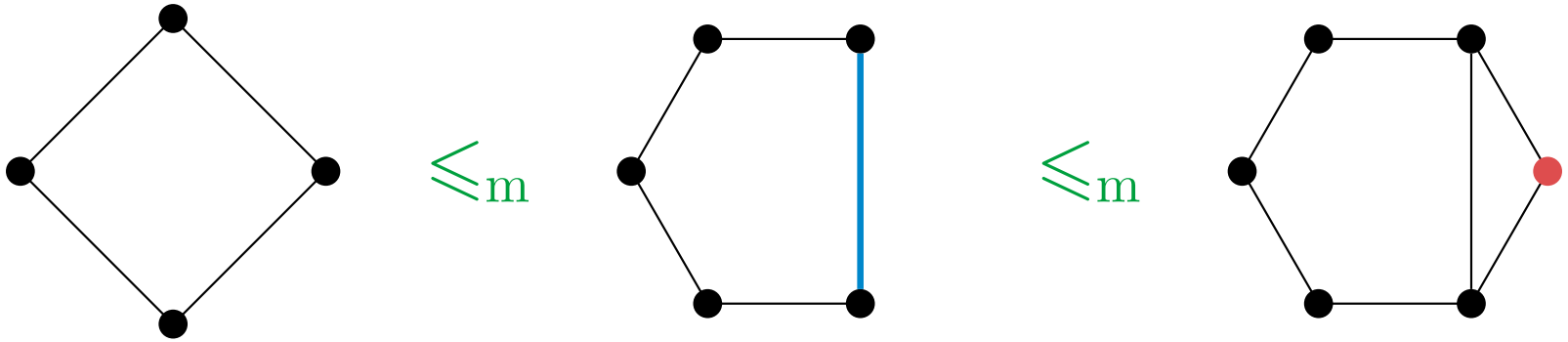
$\mathcal{F}$-Immersion-Deletion

Input: a graph G and an integer k;
Question: can I remove in G k edges to get an $\mathcal{F}$-immersion-free graph?

Many natural problems are deletion problems: Vertex Cover, Feedback Vertex Set, Feedback Arc Set, etc..

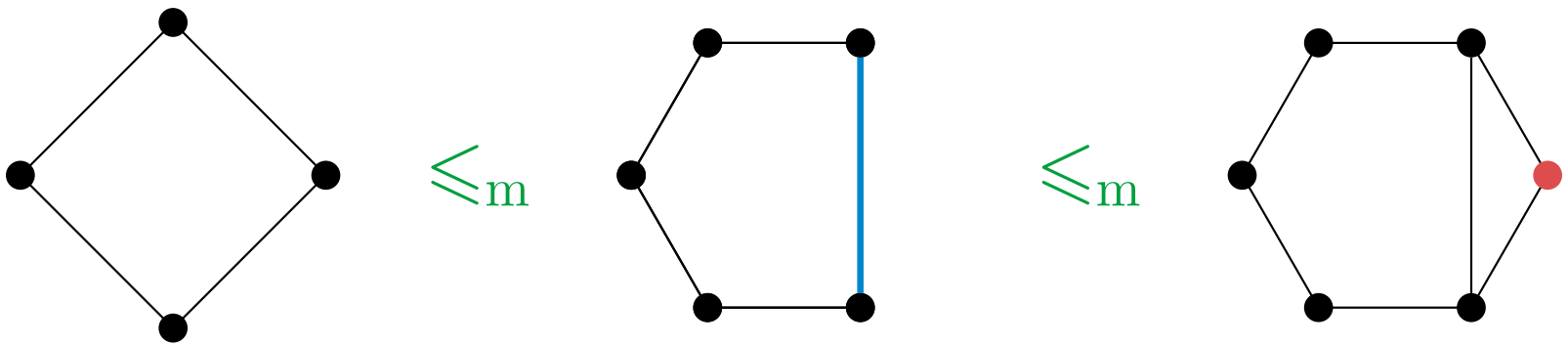
Minors and immersions

H is a *minor* of G if it can be obtained by deleting vertices or edges and contracting edges.

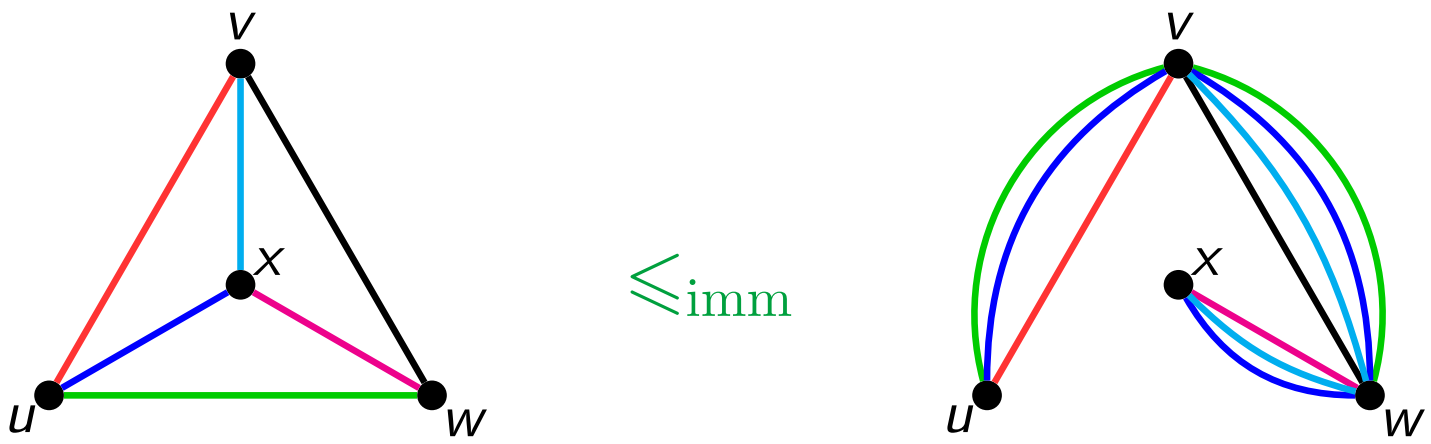


Minors and immersions

H is a *minor* of G if it can be obtained by deleting vertices or edges and contracting edges.

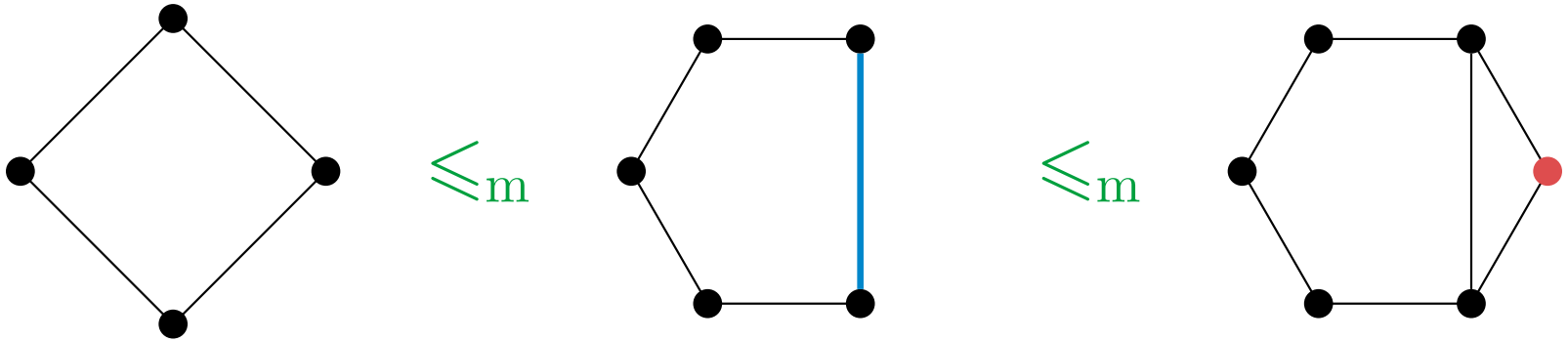


H is an *immersion* of G if it can be obtained by deleting vertices or edges and lifting edges.

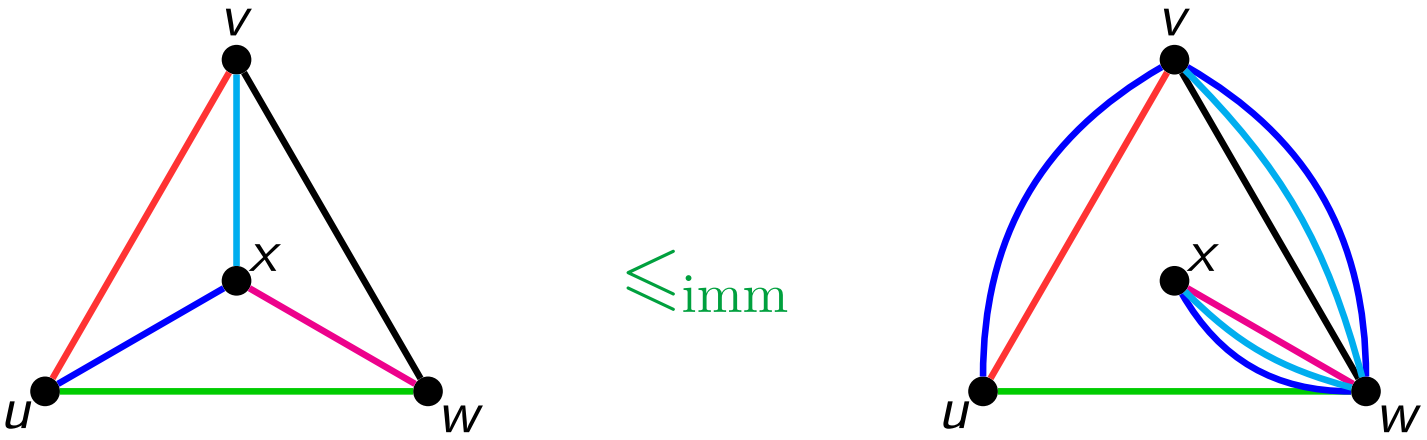


Minors and immersions

H is a *minor* of G if it can be obtained by deleting vertices or edges and contracting edges.

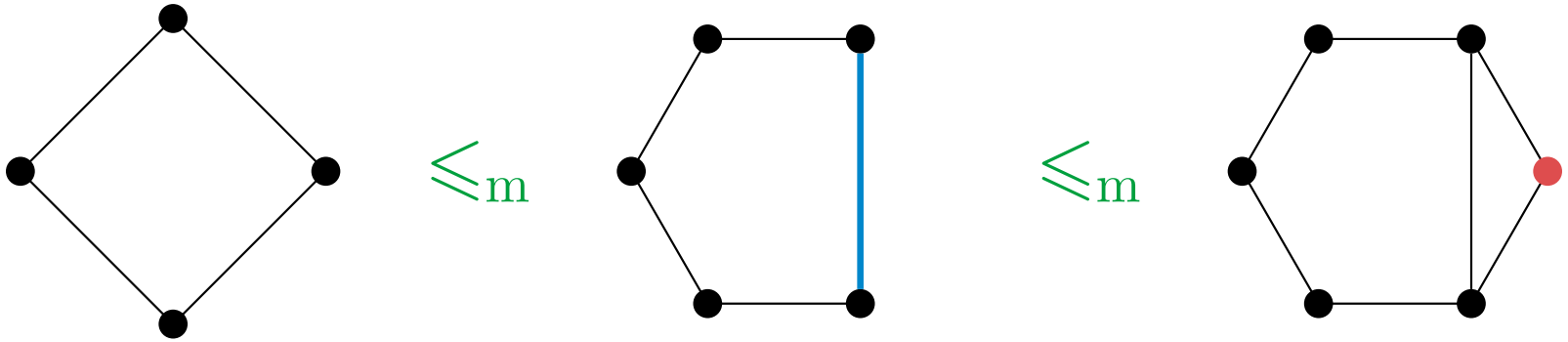


H is an *immersion* of G if it can be obtained by deleting vertices or edges and lifting edges.

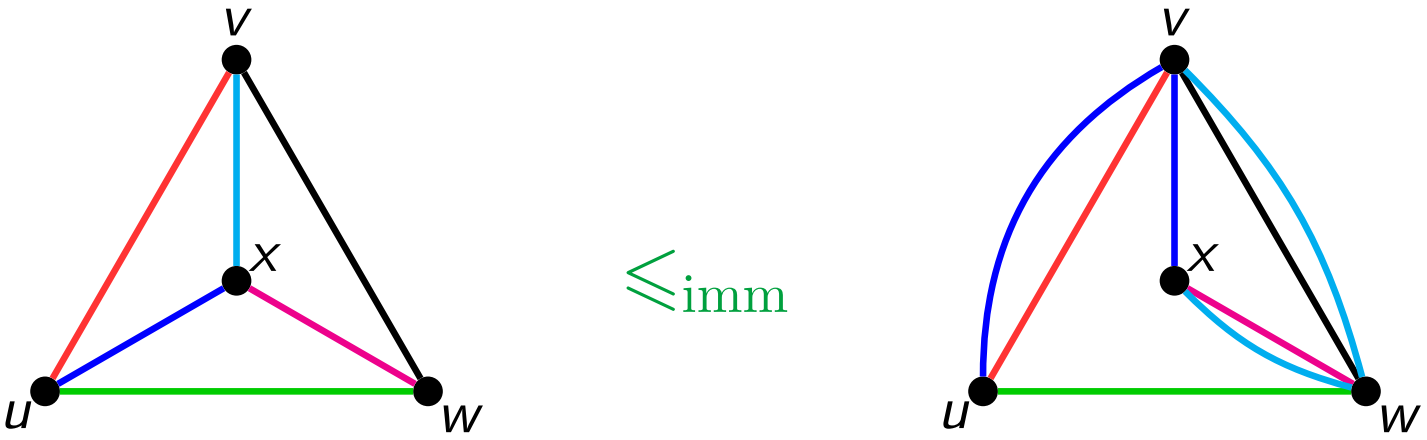


Minors and immersions

H is a *minor* of G if it can be obtained by deleting vertices or edges and contracting edges.

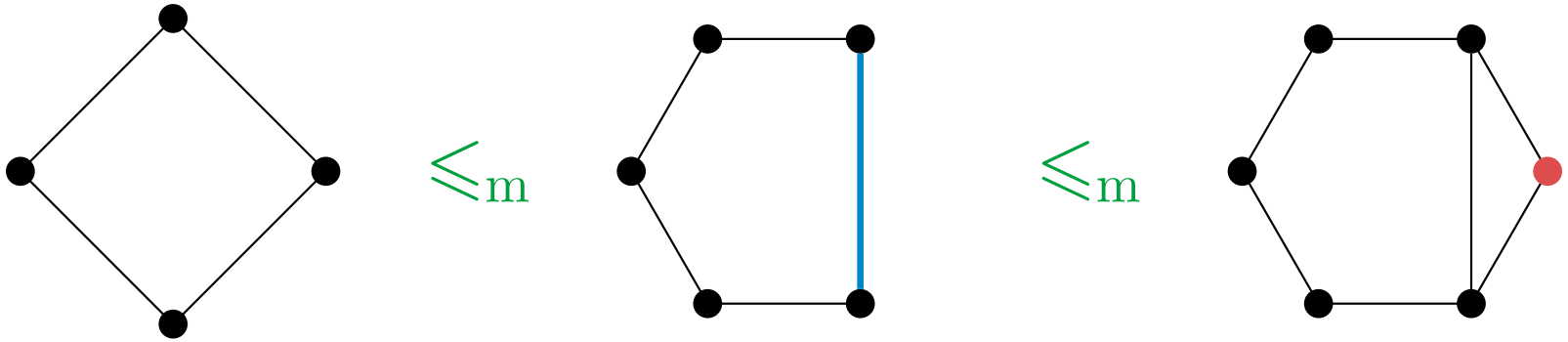


H is an *immersion* of G if it can be obtained by deleting vertices or edges and lifting edges.

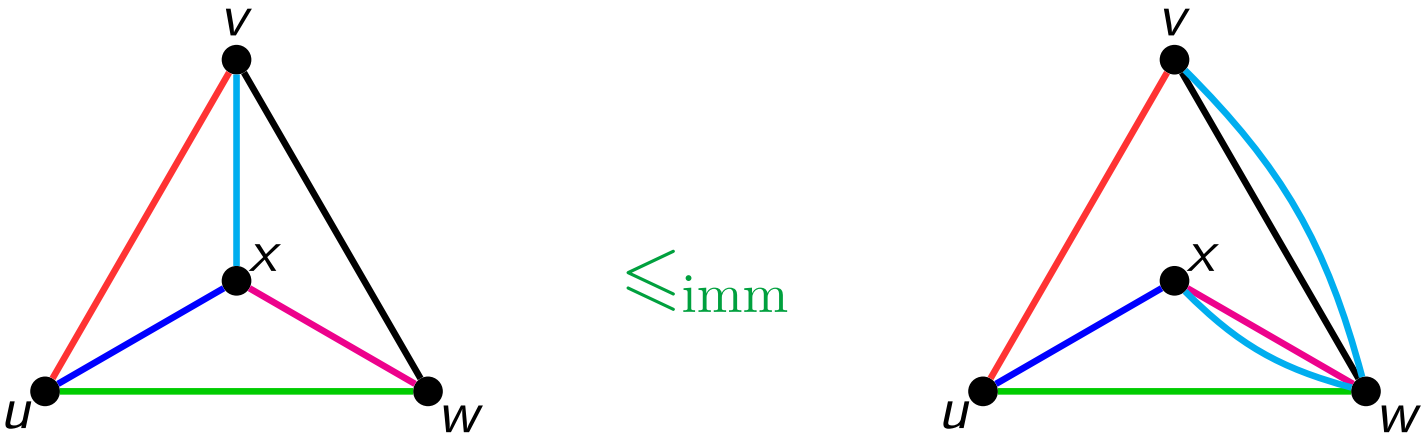


Minors and immersions

H is a *minor* of G if it can be obtained by deleting vertices or edges and contracting edges.

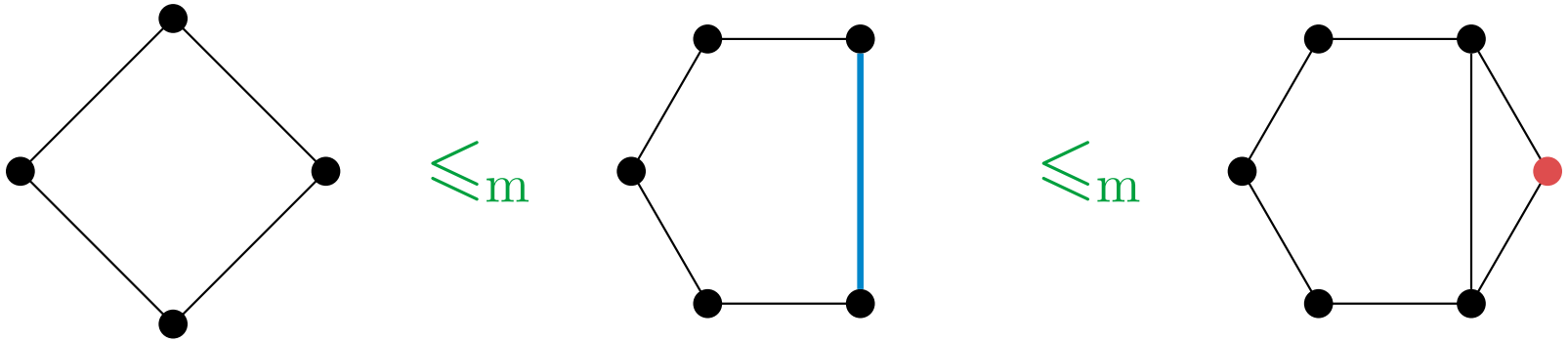


H is an *immersion* of G if it can be obtained by deleting vertices or edges and lifting edges.

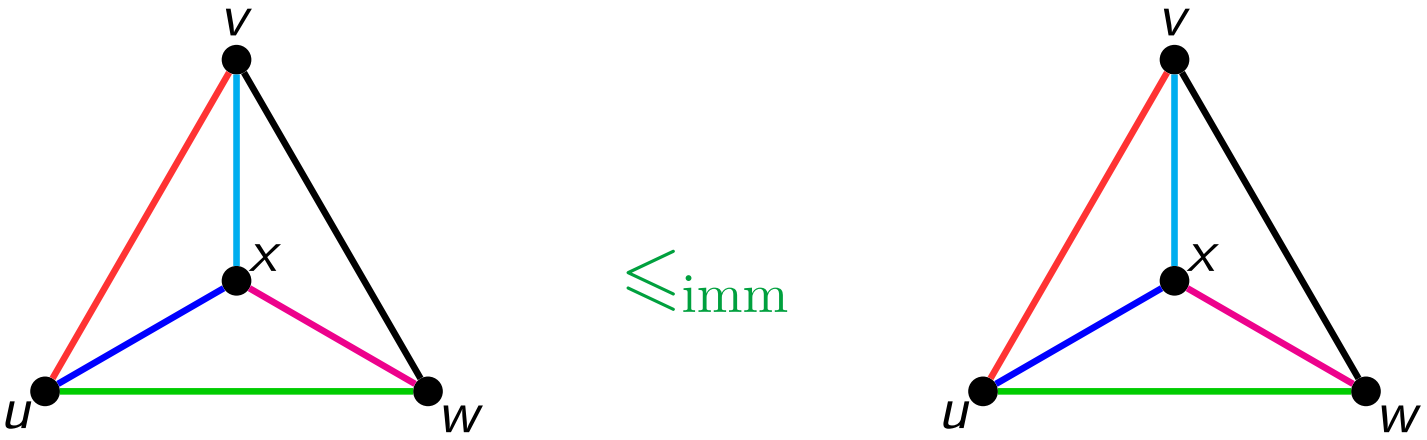


Minors and immersions

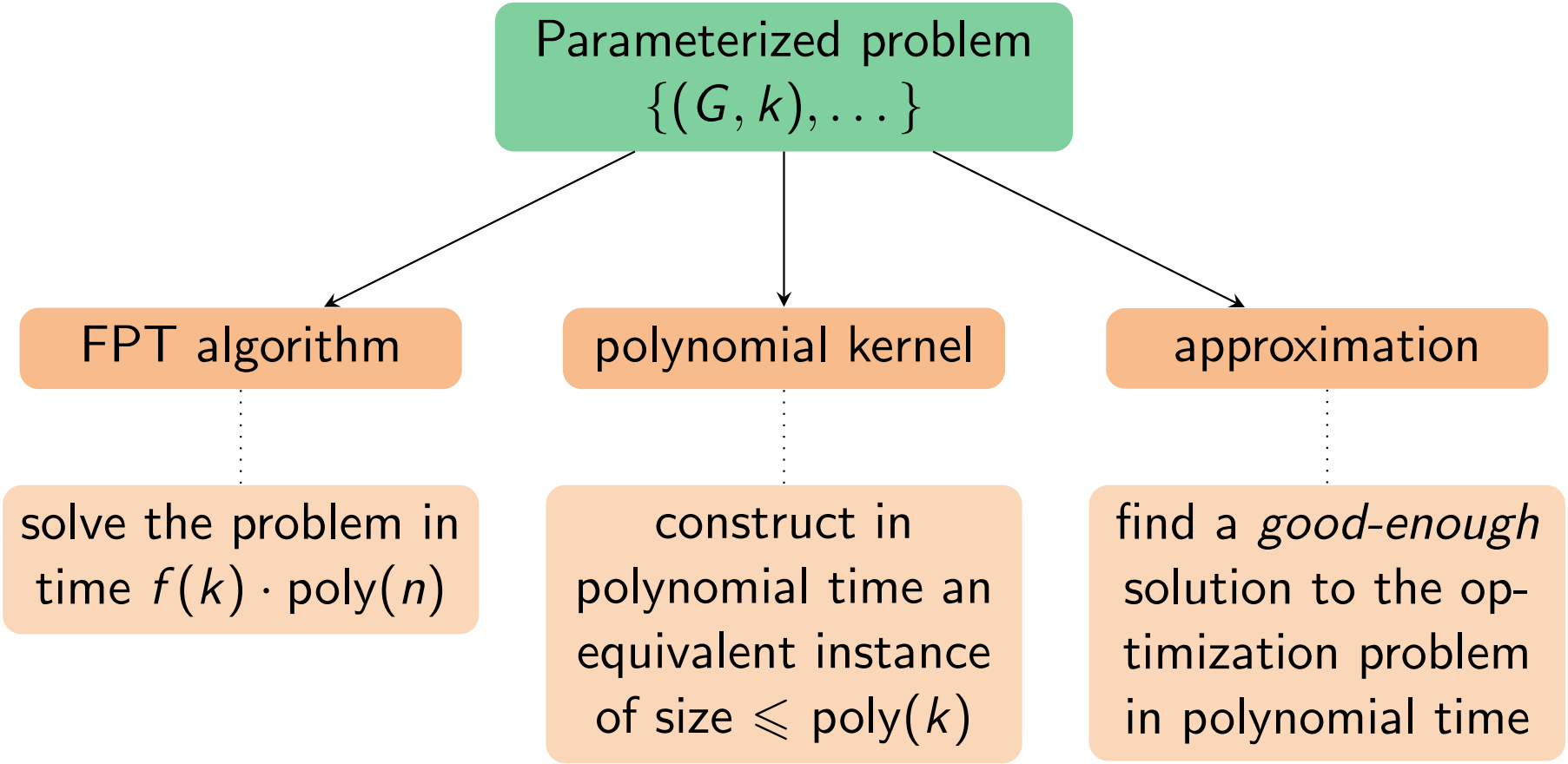
H is a *minor* of G if it can be obtained by deleting vertices or edges and contracting edges.



H is an *immersion* of G if it can be obtained by deleting vertices or edges and lifting edges.



The targets


Parameterized problem
$\{(G, k), \dots\}$
FPT algorithm
polynomial kernel
approximation
solve the problem in time $f(k) \cdot \mathrm{poly}(n)$
construct in polynomial time an equivalent instance of size $\leqslant \mathrm{poly}(k)$
find a *good-enough* solution to the optimization problem in polynomial time

Meta-theorems on deletion problems

$\mathcal{F}$: finite family of connected graphs that contains a planar graph.

Theorem [Fomin, Lokshtanov, Misra, Saurabh, FOCS 2012]

$\mathcal{F}$-Minor-Deletion (vertex removal) admits a

- randomized constant factor approximation running in polynomial time;
- $O(k^{c_{\mathcal{F}}})$-kernel that can be computed in polynomial time;
- single exponential FPT algorithm.

Meta-theorems on deletion problems

$\mathcal{F}$: finite family of connected graphs that contains a planar subcubic graph.

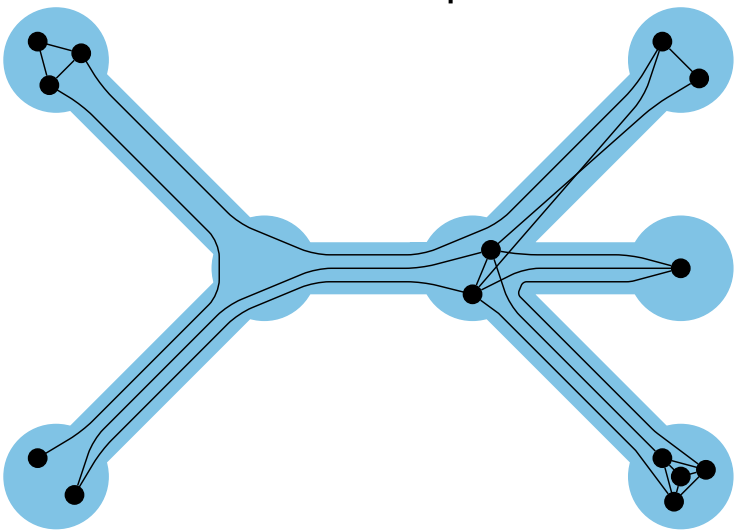
Theorem (Giannopoulou, Pilipczuk, Thilikos, R., Wrochna, 2016+)

$\mathcal{F}$-IMMERSION-DELETION (edge removal) admits a

- constant factor approximation that runs in polynomial time;
- $O(k)$-kernel that can be computed in polynomial time;
- single exponential FPT algorithm.

The tools for dealing with immersions

Tree-cut decomposition:



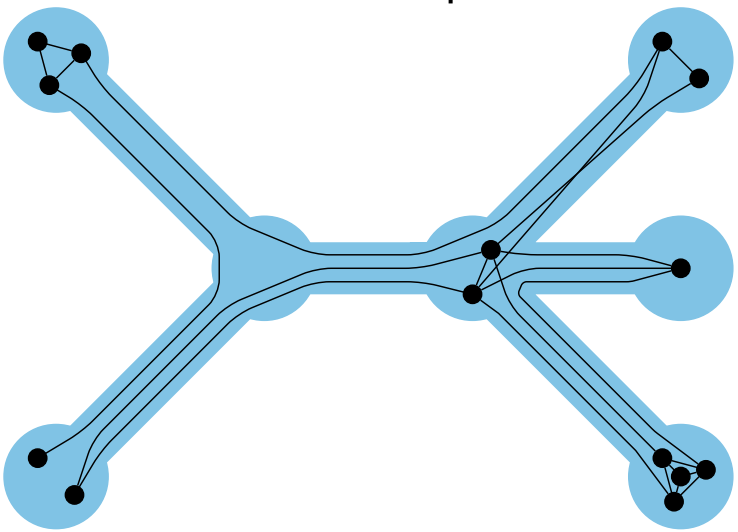
Associated parameter: **tcw**
(tree-cut width)

Small **tcw** implies:

- small bags;
- *thin* edges;
- small number of *thick* neighbors.

The tools for dealing with immersions

Tree-cut decomposition:



Associated parameter: **tcw**
(tree-cut width)

Small **tcw** implies:

- small bags;
- *thin* edges;
- small number of *thick* neighbors.

Theorem [Wollan, JCTB 2015] + [Chuzhoy STOC 2015+]

For every planar subcubic graph H on h edges,

$$H \not\leq_{\mathrm{imm}} G \quad \Rightarrow \quad \mathbf{tcw}(G) = O(h^{30}).$$

What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges

What does a YES-instance look like?

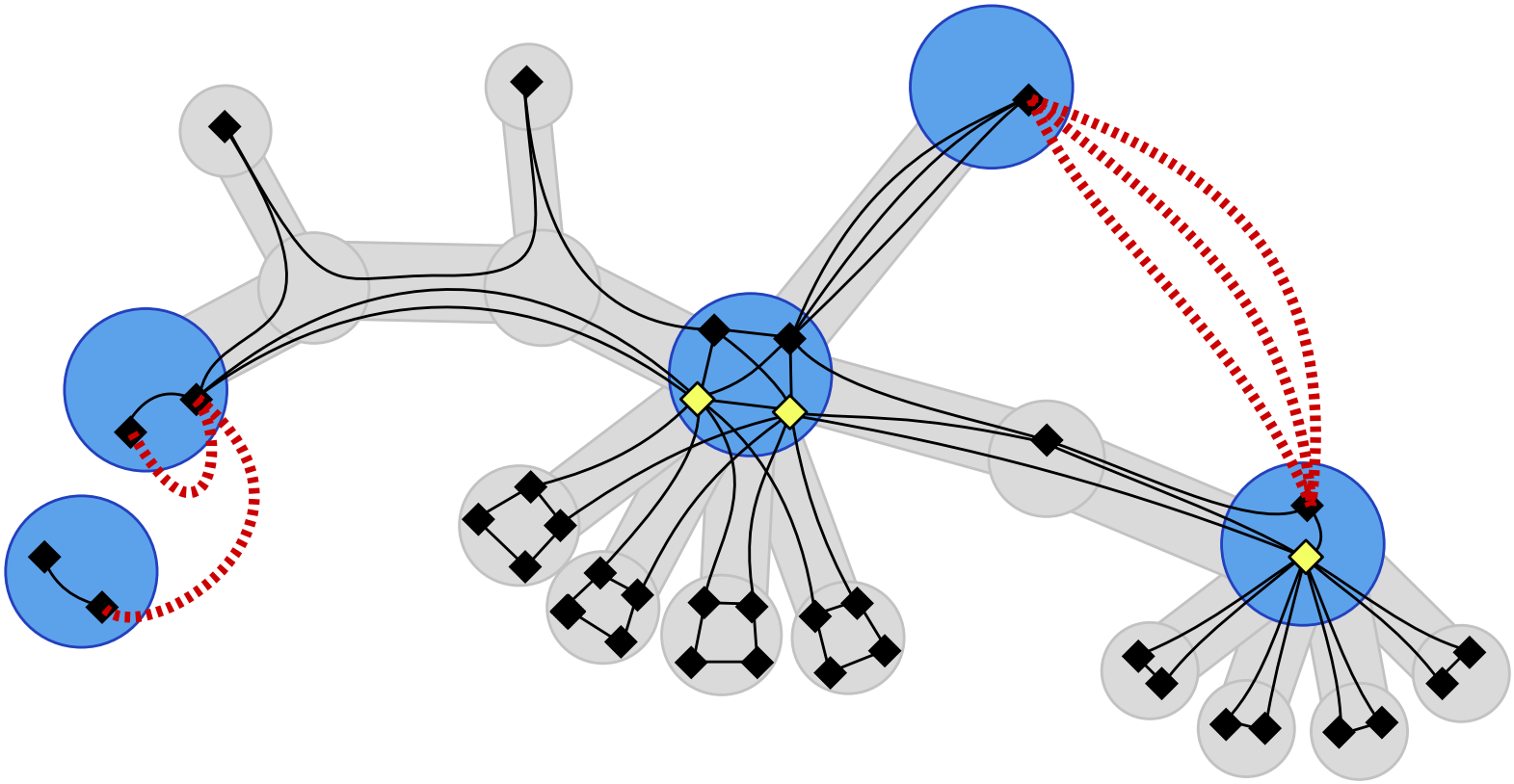
Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges
$=$ graph of small **tcw** $+$ k edges

What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

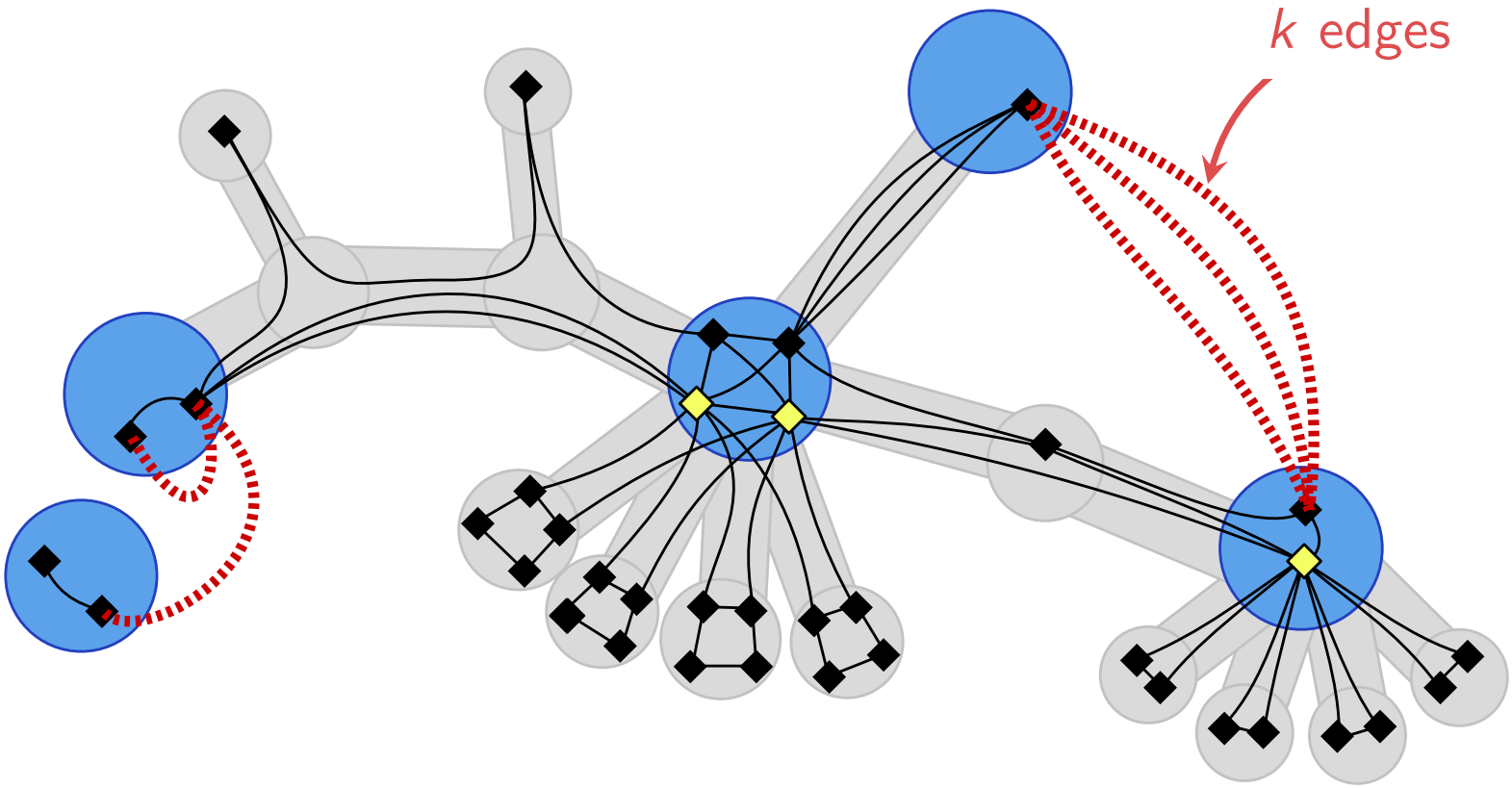
YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges
$=$ graph of small **tcw** $+$ k edges



What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

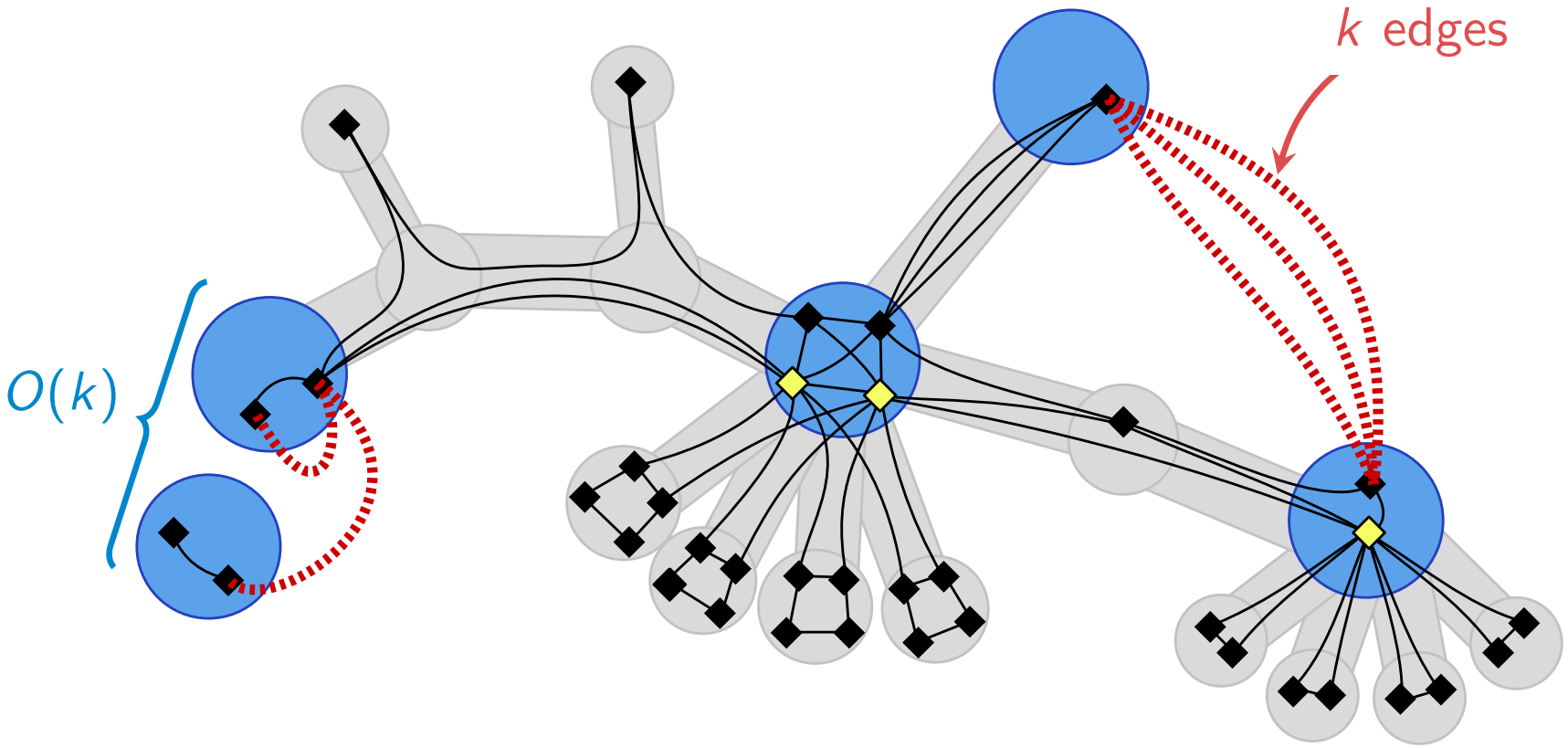
YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges
$=$ graph of small **tcw** $+$ k edges



What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

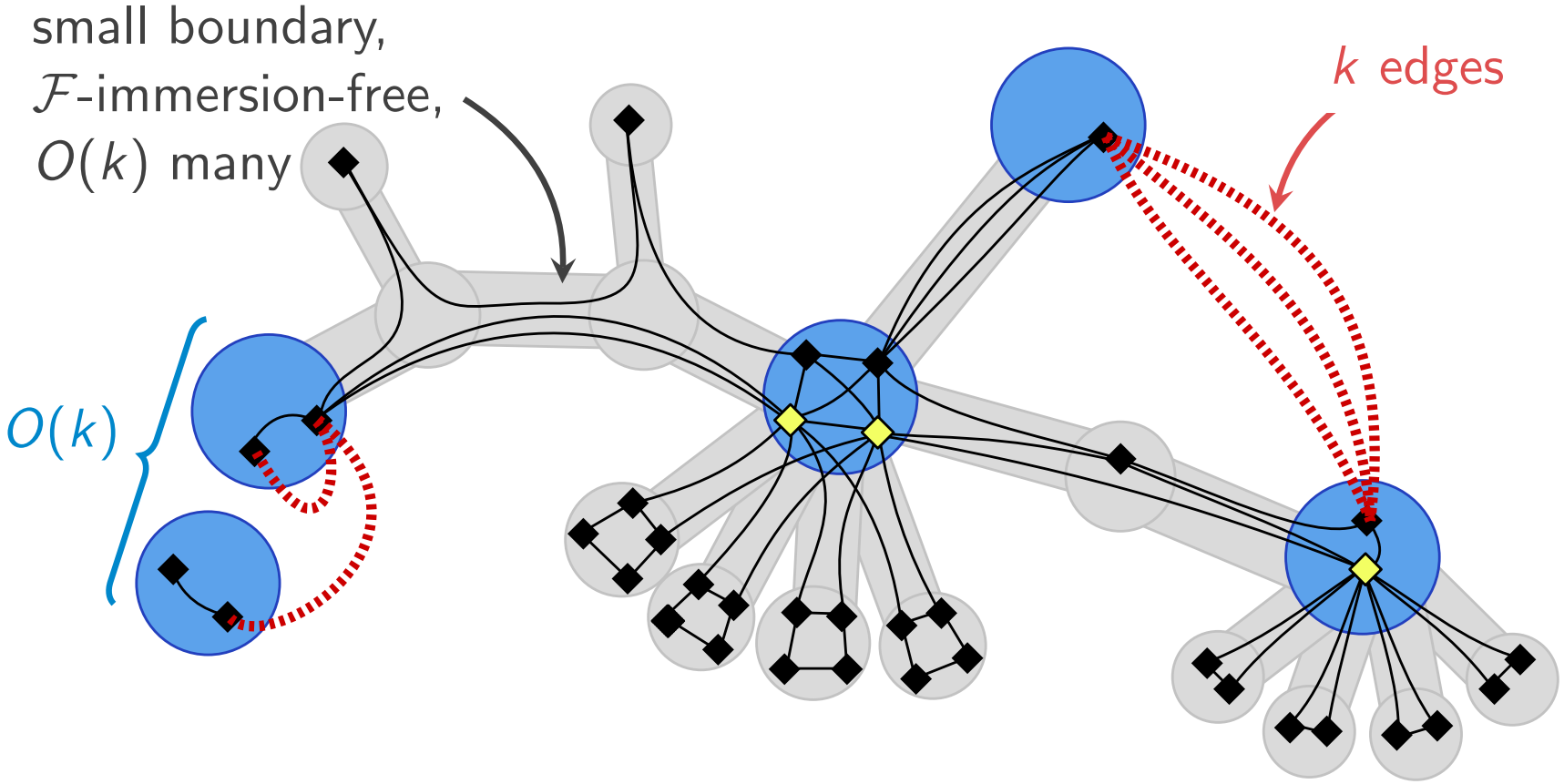
YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges
$=$ graph of small **tcw** $+$ k edges



What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

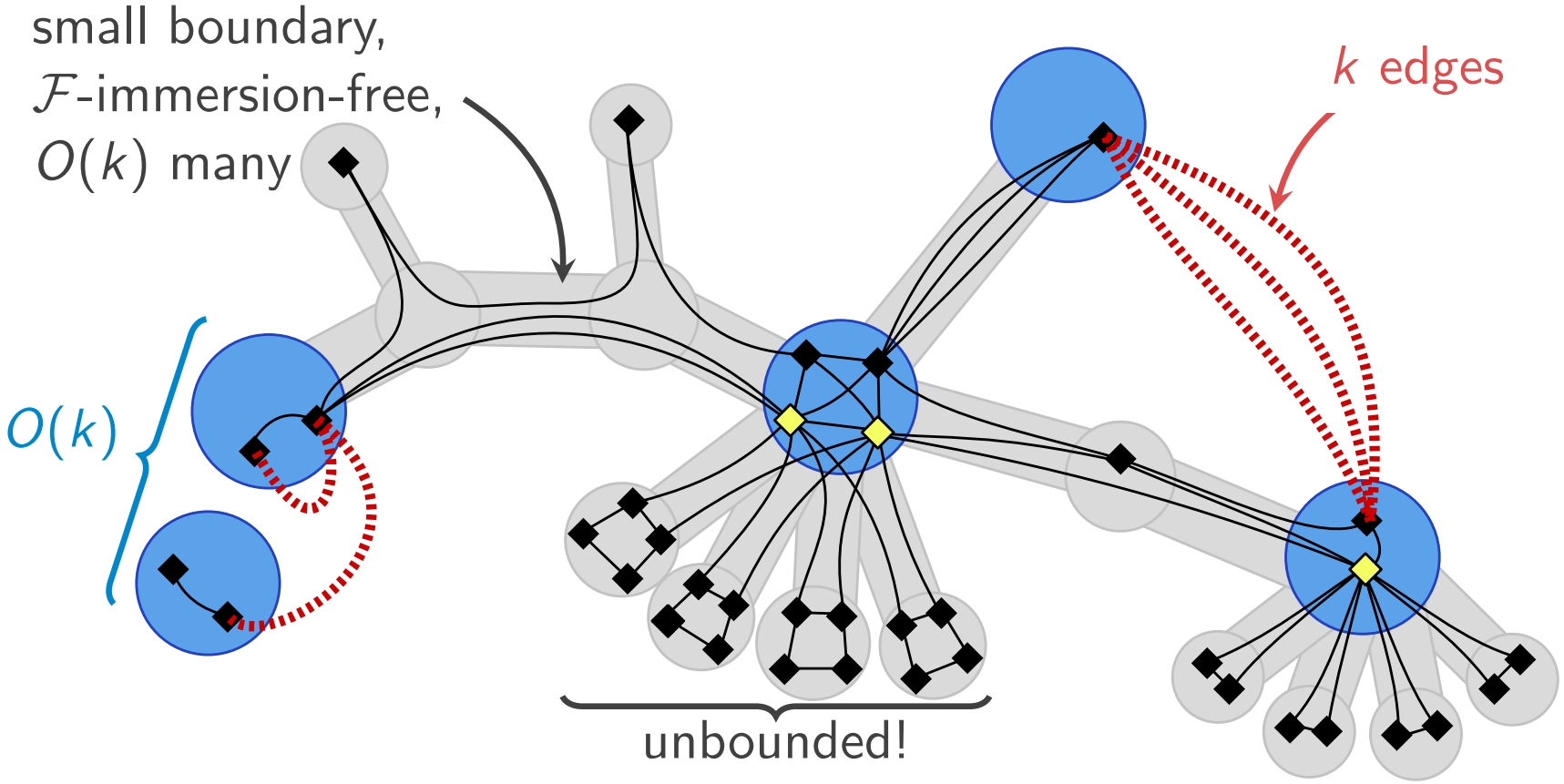
YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges
$=$ graph of small **tcw** $+$ k edges



What does a YES-instance look like?

Problem: decide if deleting k edges gives an $\mathcal{F}$-immersion-free graph.

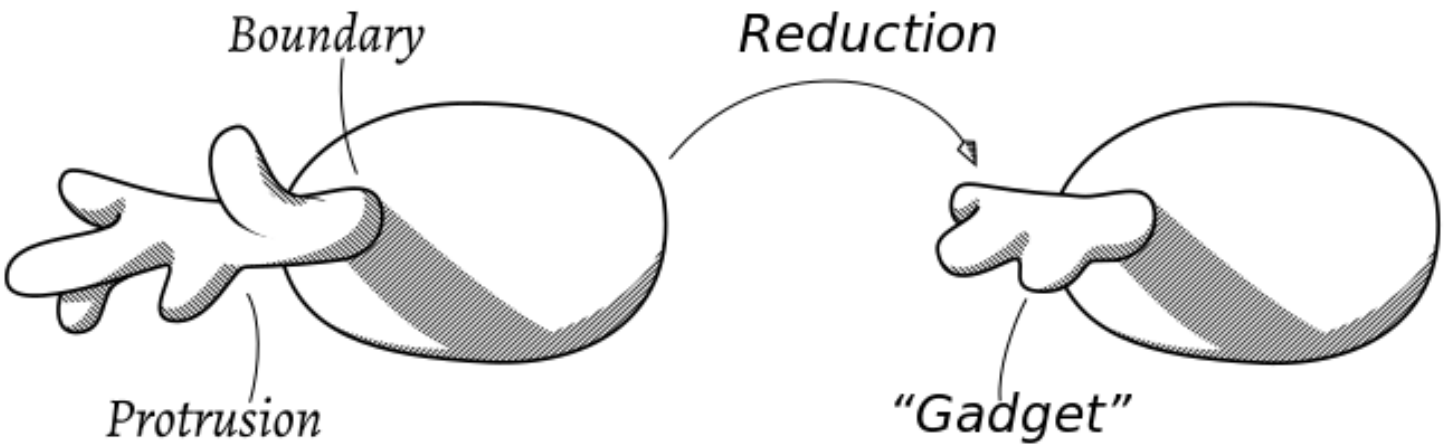
YES-instance $=$ $\mathcal{F}$-immersion-free graph $+$ k edges
$=$ graph of small **tcw** $+$ k edges



Replacing protrusions

A protrusion is part of the graph that

- has a small boundary;
- is $\mathcal{F}$-immersion-free.

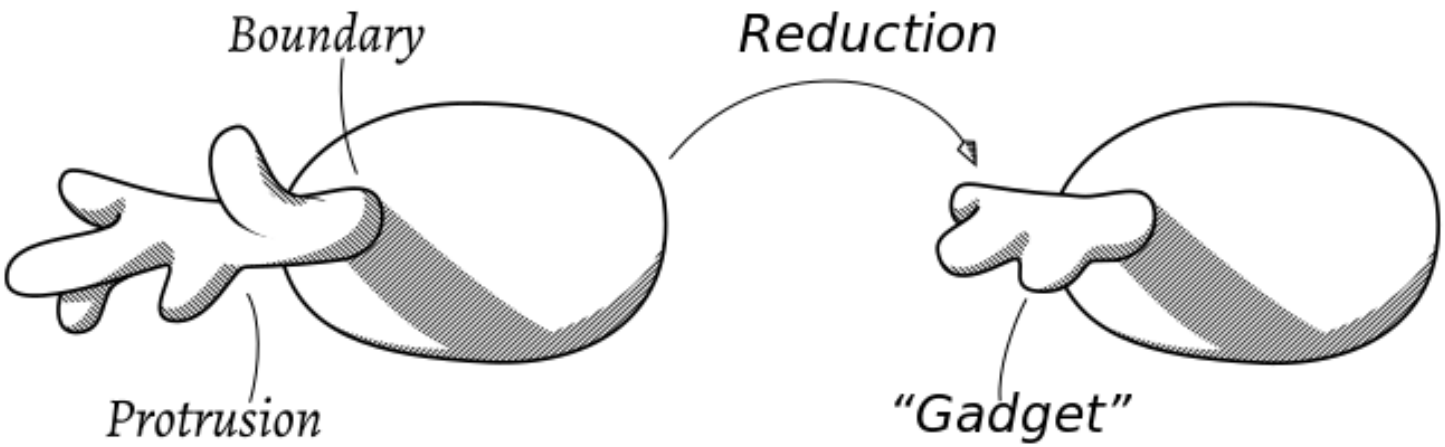


(Picture by F. Reidl)

Replacing protrusions

An edge-protrusion is part of the graph that

- has a small edge-boundary;
- is $\mathcal{F}$-immersion-free.

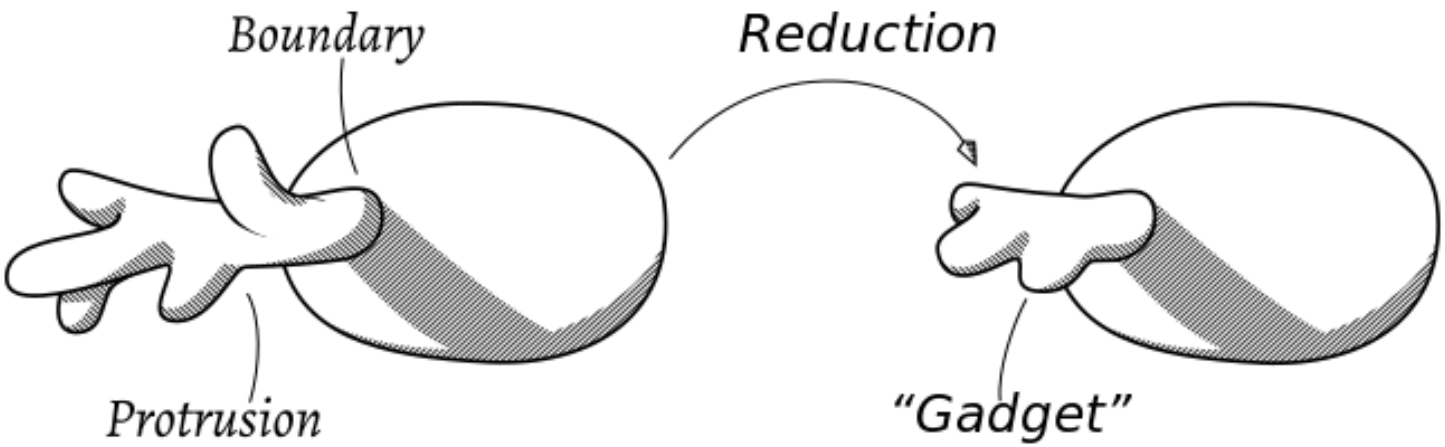


(Picture by F. Reidl)

Replacing protrusions

An edge-protrusion is part of the graph that

- has a small edge-boundary;
- is $\mathcal{F}$-immersion-free.



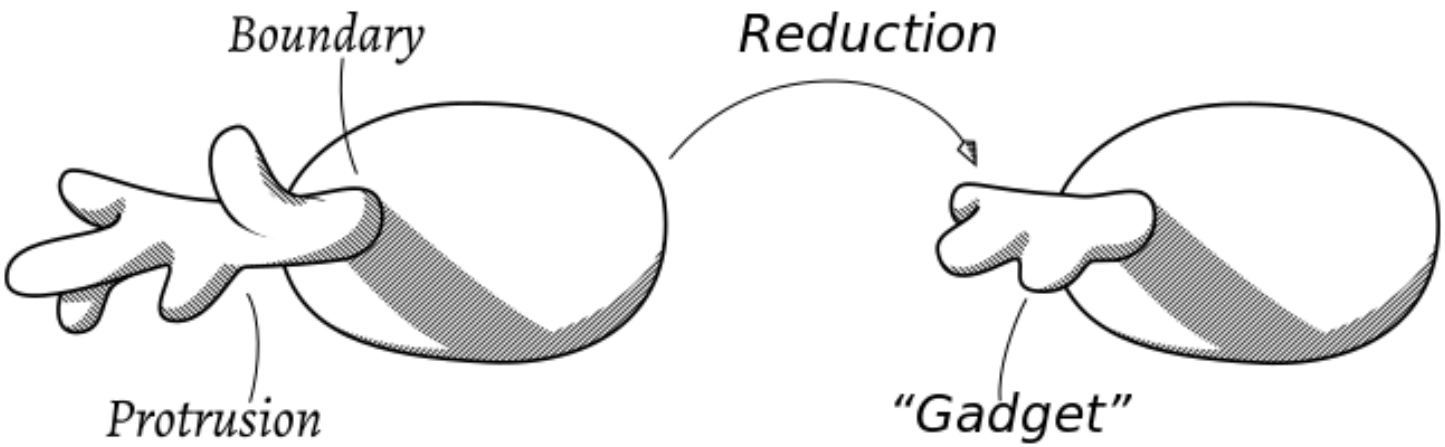
1. define an equivalence relation on protrusions;

(Picture by F. Reidl)

Replacing protrusions

An edge-protrusion is part of the graph that

- has a small edge-boundary;
- is $\mathcal{F}$-immersion-free.



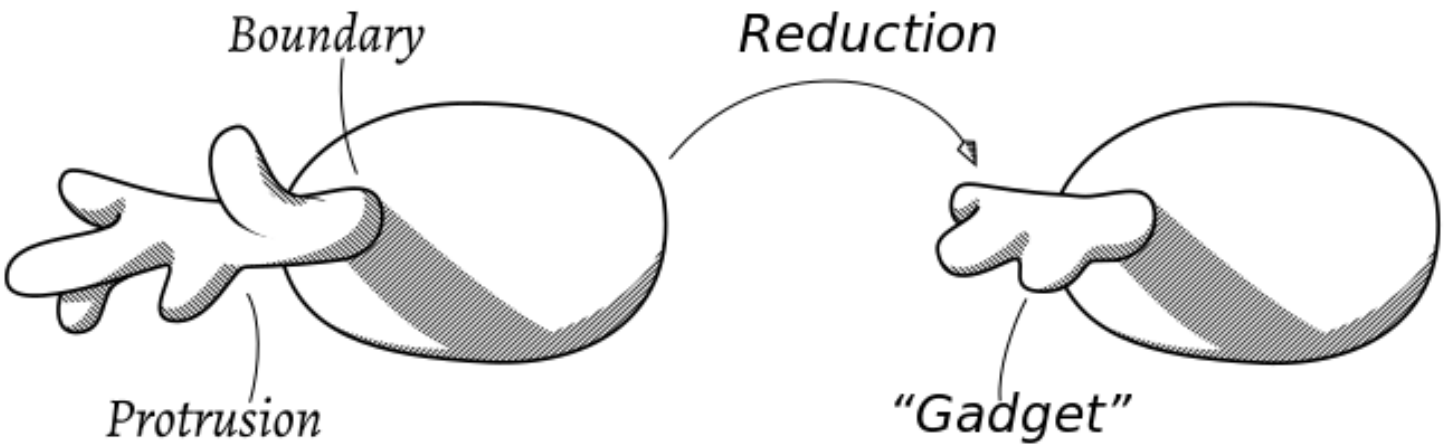
1. define an equivalence relation on protrusions;
2. show that there is a finite number of equivalence classes;

(Picture by F. Reidl)

Replacing protrusions

An edge-protrusion is part of the graph that

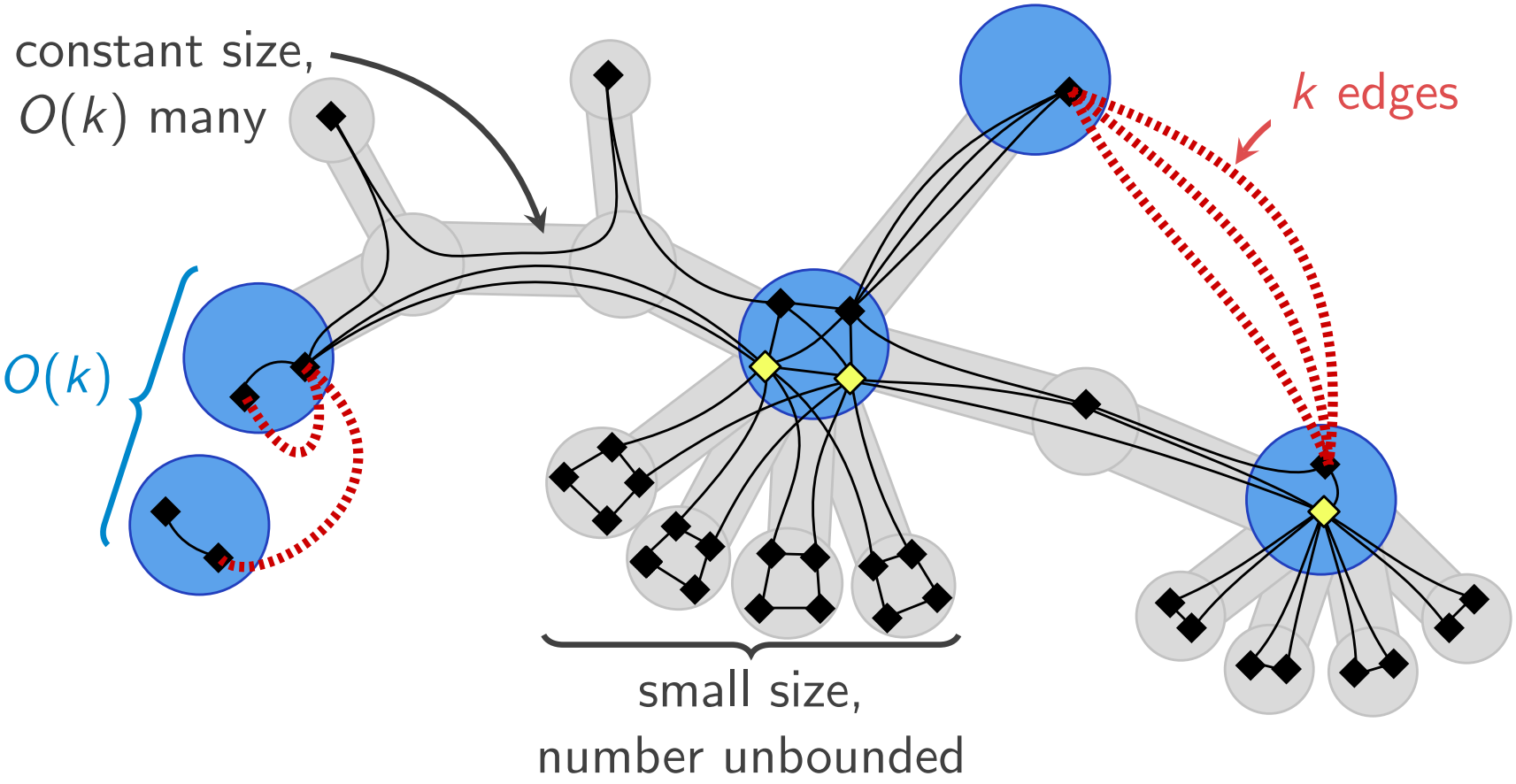
- has a small edge-boundary;
- is $\mathcal{F}$-immersion-free.



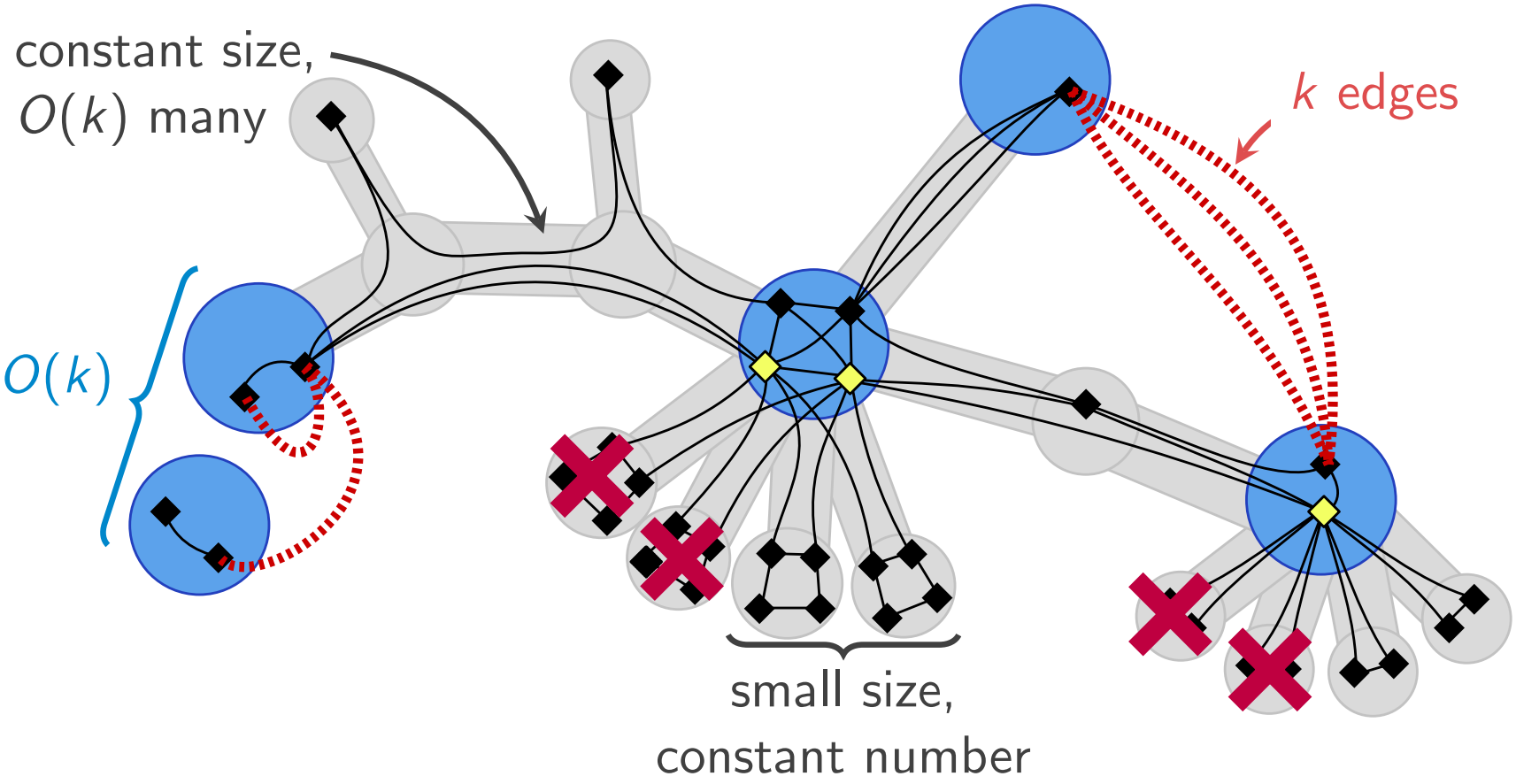
1. define an equivalence relation on protrusions;
2. show that there is a finite number of equivalence classes;
3. replace large protrusions by smaller equivalent ones.

(Picture by F. Reidl)

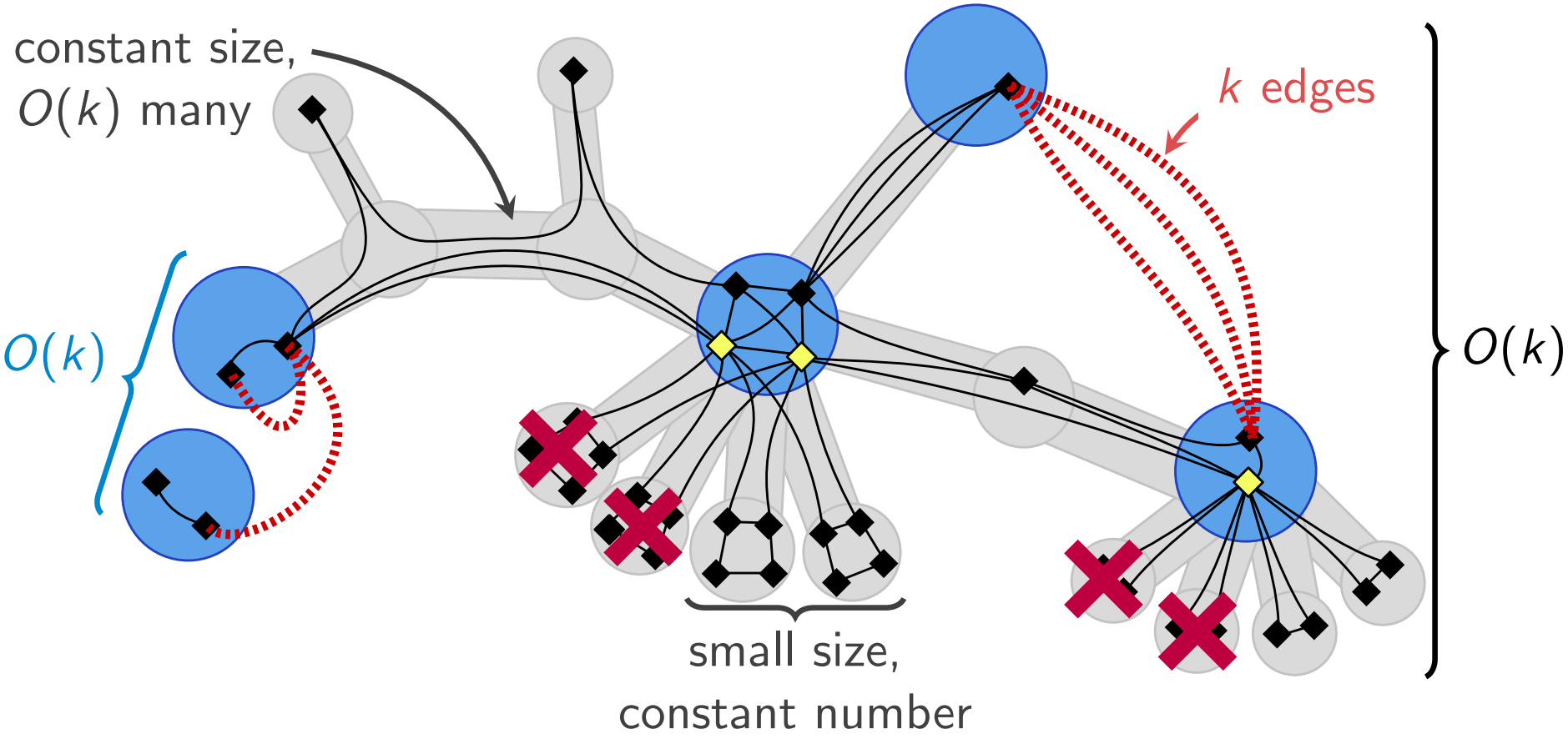
What does a reduced YES-instance look like?



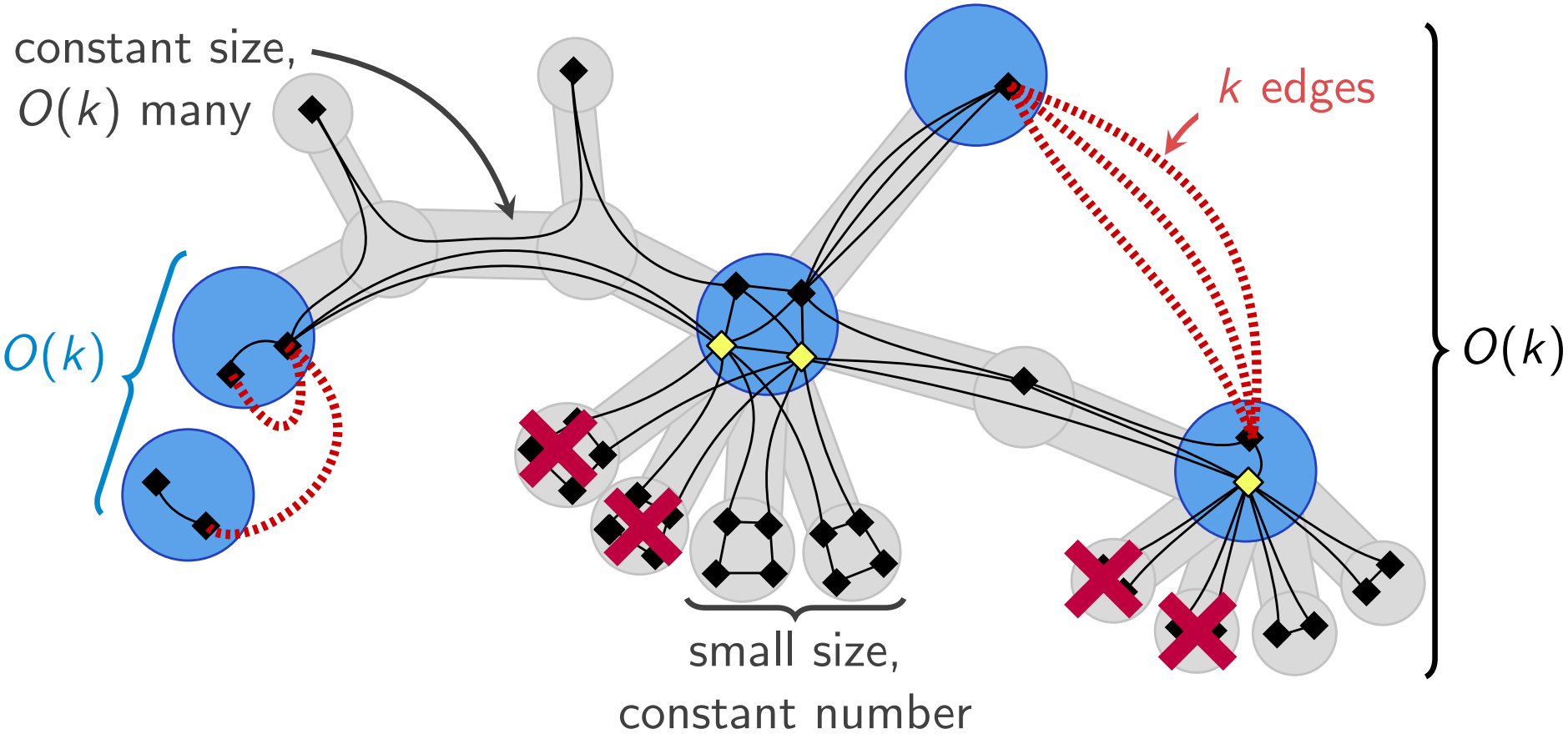
What does a reduced YES-instance look like?



What does a reduced YES-instance look like?



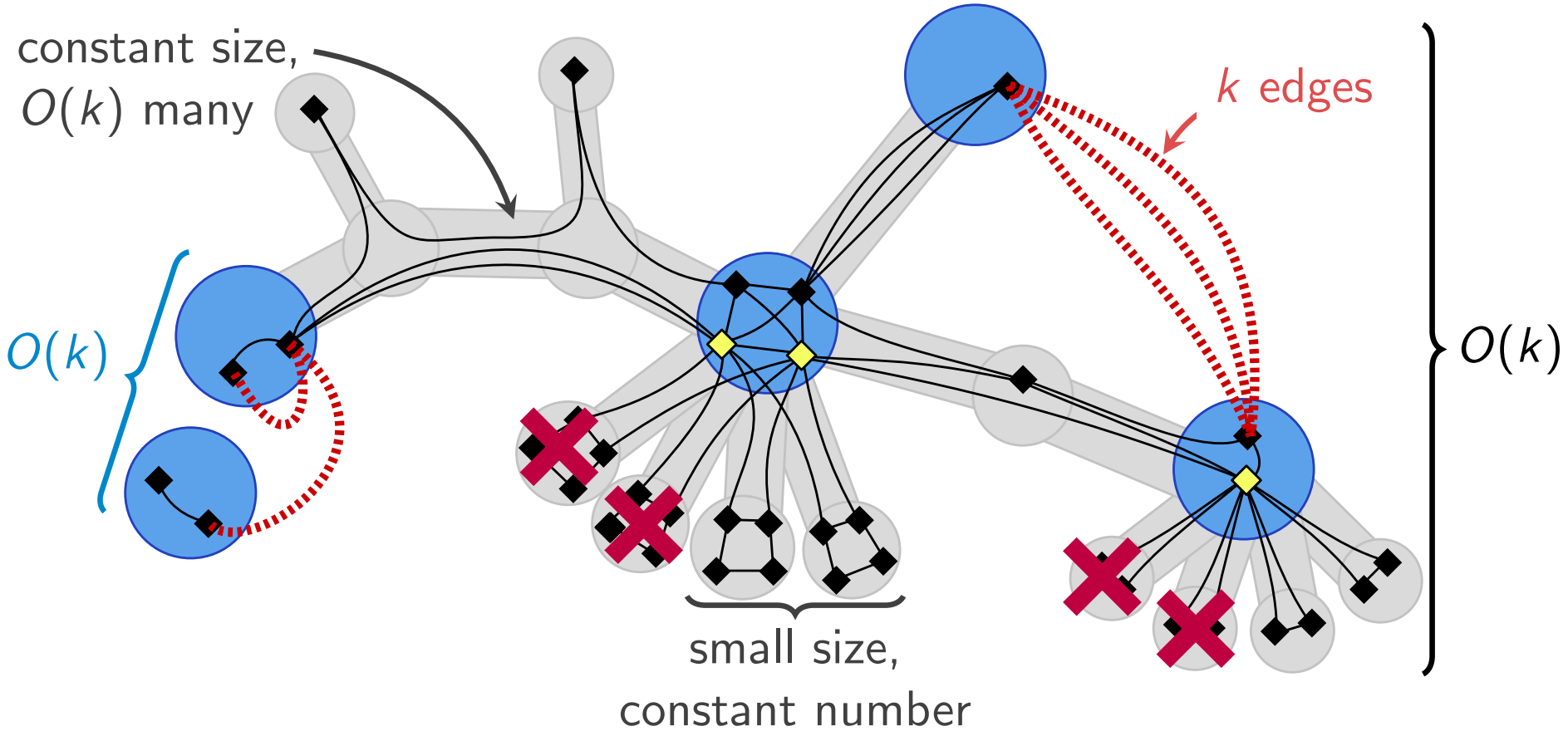
What does a reduced YES-instance look like?



Lemma

Removing these $O(k)$ edges from the graph strictly decreases the edge-deletion-distance to $\mathcal{F}$-immersion-free.

What does a reduced YES-instance look like?

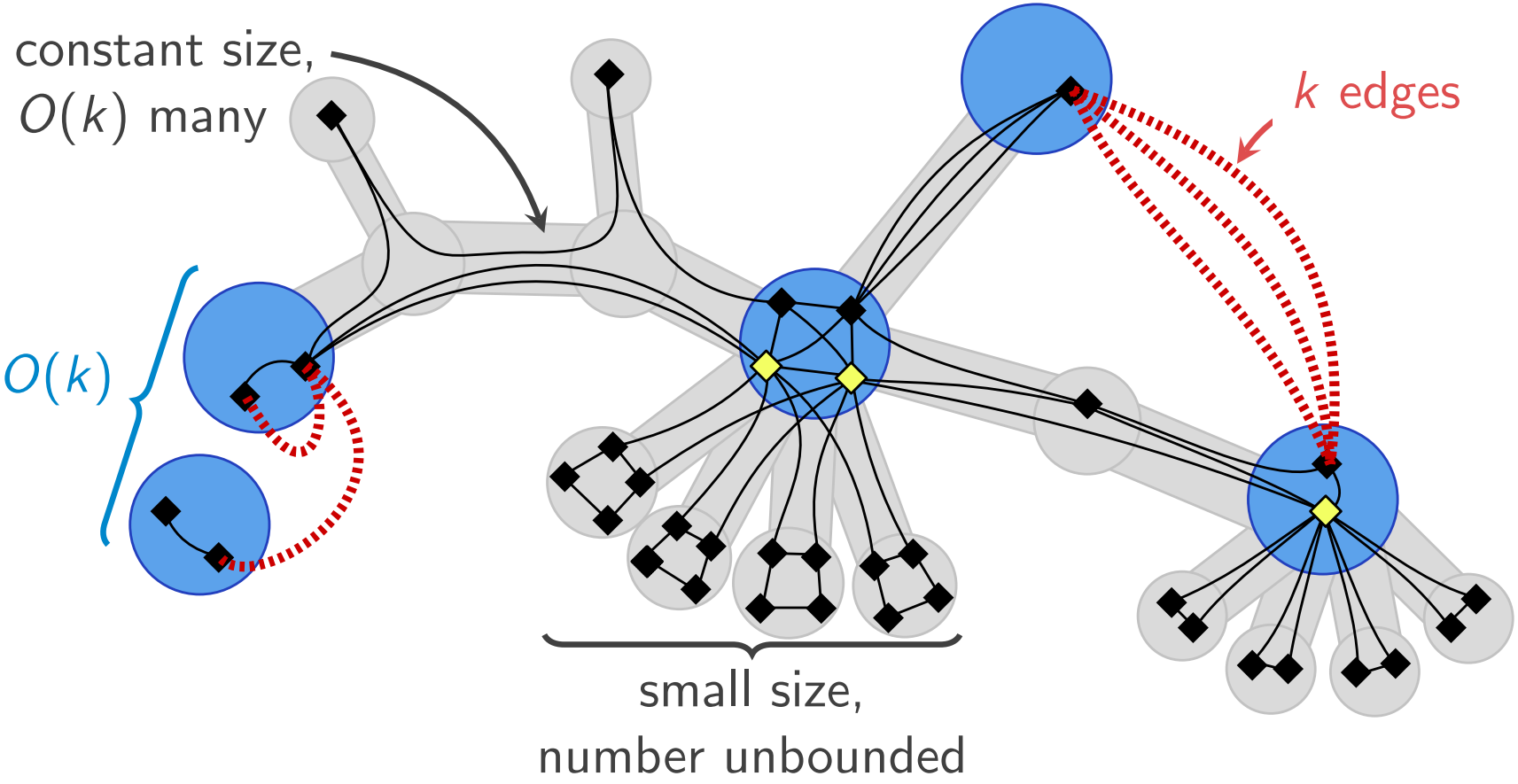


Lemma

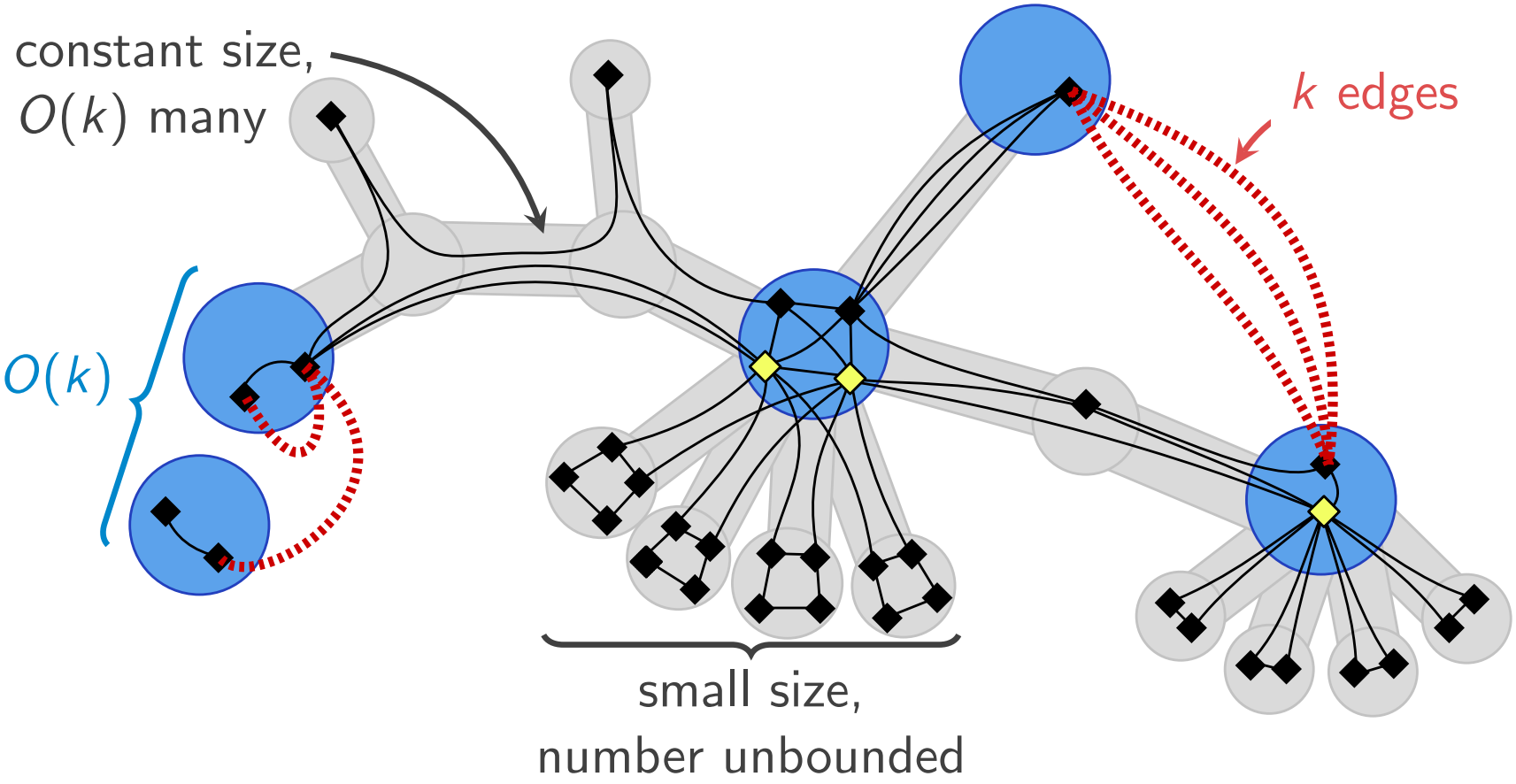
Removing these $O(k)$ edges from the graph strictly decreases the edge-deletion-distance to $\mathcal{F}$-immersion-free.

$\rightsquigarrow$ *iterate and get an approximation.*

The kernel

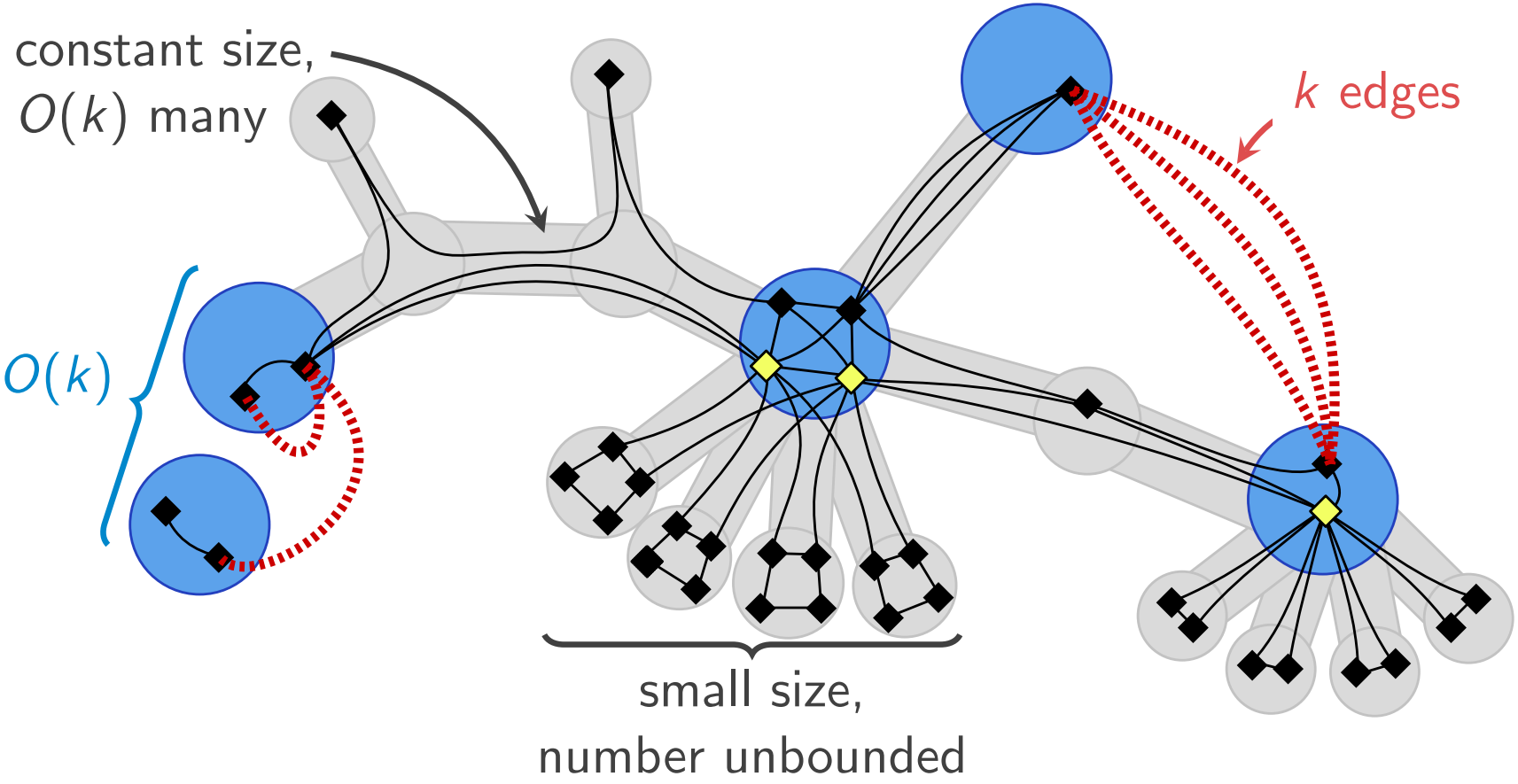

constant size,
$O(k)$ many
k edges
$O(k)$
small size,
number unbounded

The kernel



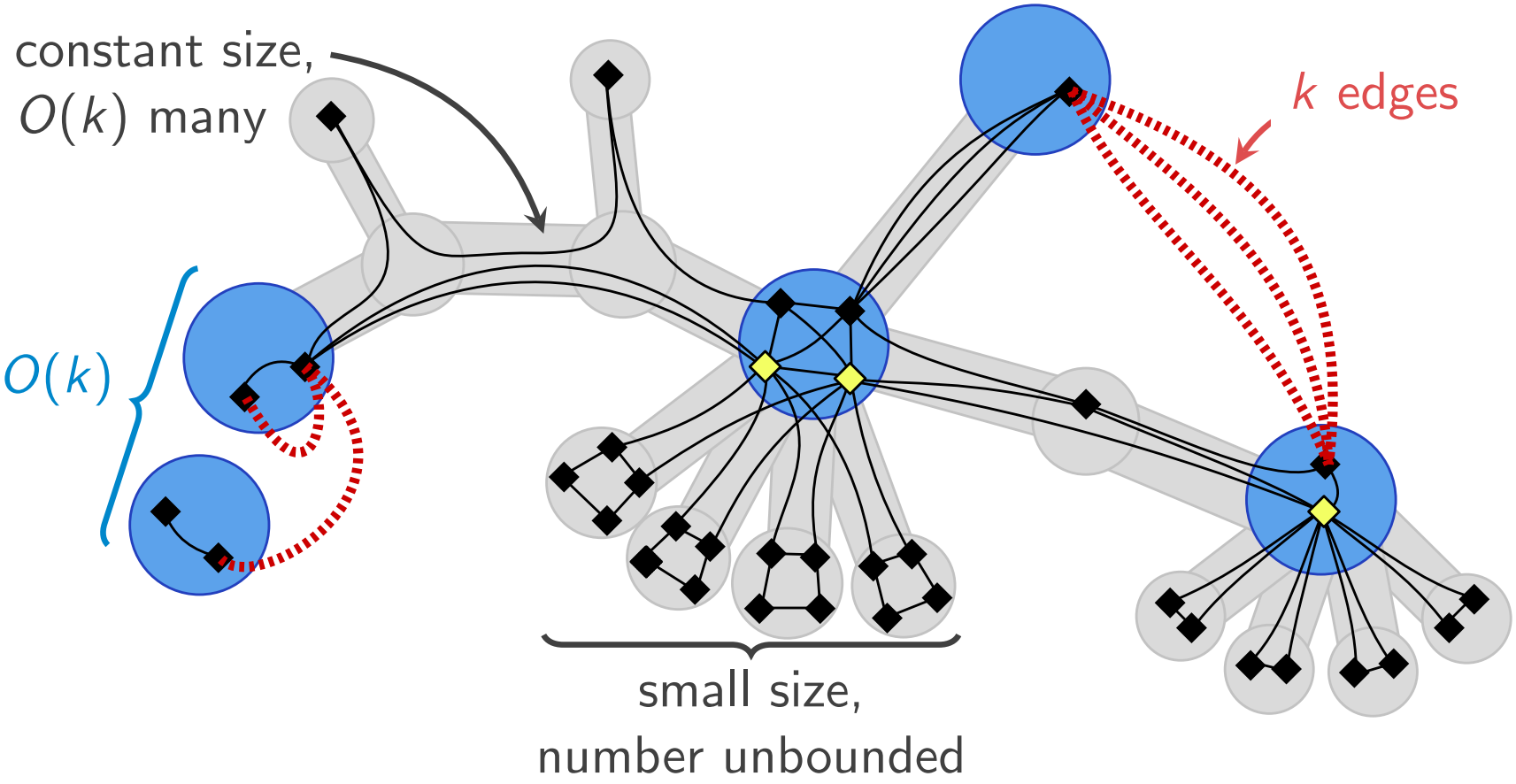
1. compute an approximate solution F;

The kernel



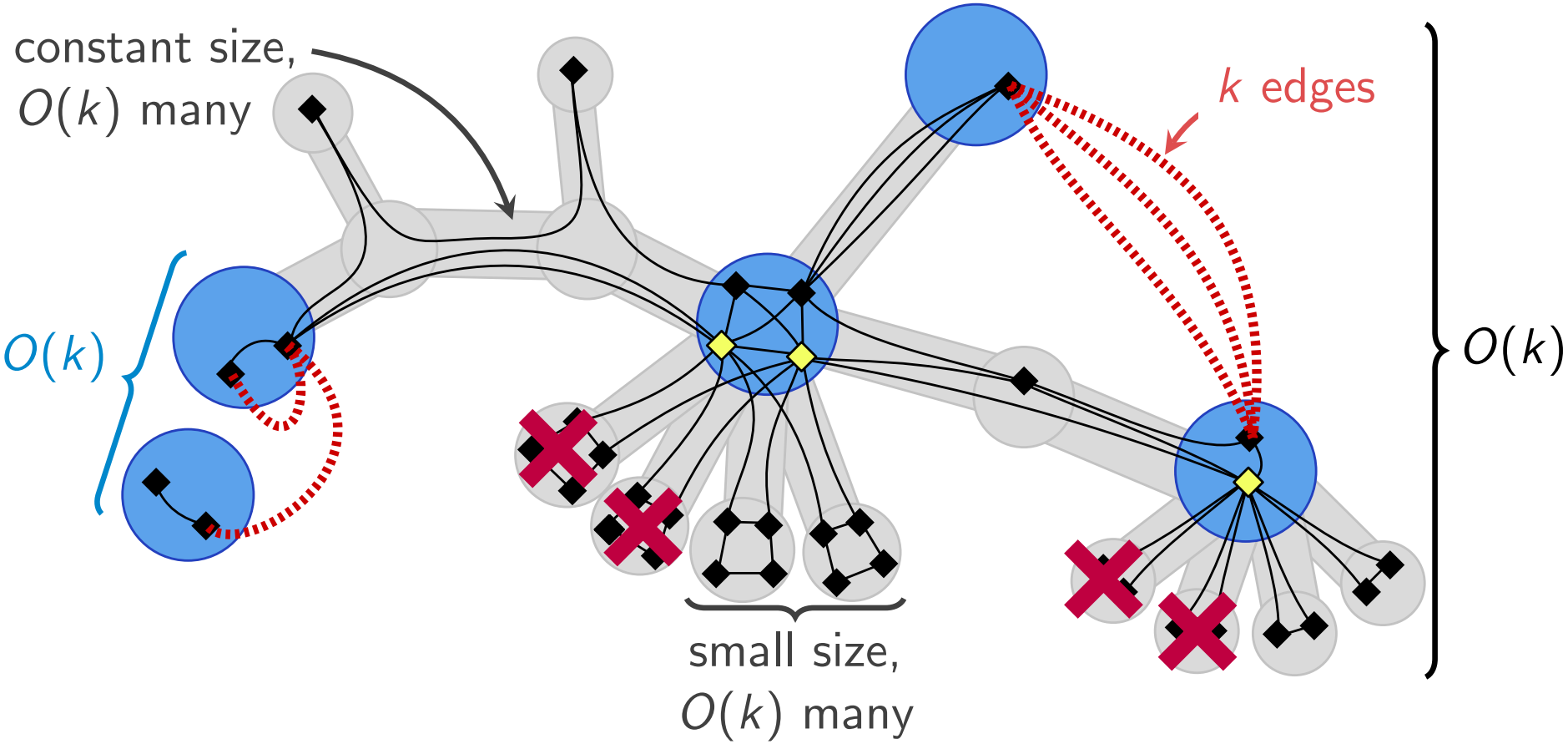
1. compute an approximate solution F;
2. count for every bouquet how many edges *affect* this bouquet;

The kernel



1. compute an approximate solution F;
2. count for every bouquet how many edges *affect* this bouquet;
3. only keep in each bouquet a number of graphs that is linear in the number of edges affecting it.

The kernel



1. compute an approximate solution F;
2. count for every bouquet how many edges *affect* this bouquet;
3. only keep in each bouquet a number of graphs that is linear in the number of edges affecting it. $\rightsquigarrow$ kernel

Application to obstructions

$\mathcal{G} = \mathcal{F}$-immersion-free graphs
$\mathcal{G}_k =$ graphs where the deletion of some k edges gives a graph of $\mathcal{G}$

Application to obstructions

$\mathcal{G} = \mathcal{F}$-immersion-free graphs
$\mathcal{G}_k =$ graphs where the deletion of some k edges gives a graph of $\mathcal{G}$

- $\mathcal{G}_k$ is immersion closed;
- $\Rightarrow$ it has finitely many obstructions.

Application to obstructions

$\mathcal{G} = \mathcal{F}$-immersion-free graphs
$\mathcal{G}_k =$ graphs where the deletion of some k edges gives a graph of $\mathcal{G}$

- $\mathcal{G}_k$ is immersion closed;
- $\Rightarrow$ it has finitely many obstructions.

Theorem (Giannopoulou, Pilipczuk, Thilikos, R., Wrochna, 2016+)

Every immersion-minimal graph that does not belong to $\mathcal{G}_k$ has at most $c_{\mathcal{F}} \cdot k$ edges.

Application to obstructions

$\mathcal{G} = \mathcal{F}$-immersion-free graphs
$\mathcal{G}_k =$ graphs where the deletion of some k edges gives a graph of $\mathcal{G}$

- $\mathcal{G}_k$ is immersion closed;
- $\Rightarrow$ it has finitely many obstructions.

Theorem (Giannopoulou, Pilipczuk, Thilikos, R., Wrochna, 2016+)

Every immersion-minimal graph that does not belong to $\mathcal{G}_k$ has at most $c_{\mathcal{F}} \cdot k$ edges.

Proof: the kernel we construct is an immersion of the original graph.

Application to immersion closed parameters

Let **p** be a parameter that is:

- closed under immersion and disjoint union;
- large on walls.

Application to immersion closed parameters

Let **p** be a parameter that is:

- closed under immersion and disjoint union;
- large on walls.

For instance: cutwidth, carving-width, tree-cut width, etc..

Application to immersion closed parameters

Let **p** be a parameter that is:

- closed under immersion and disjoint union;
- large on walls.

For instance: cutwidth, carving-width, tree-cut width, etc..

Theorem (Giannopoulou, Pilipczuk, Thilikos, R., Wrochna, 2016+)

For every $r \in \mathbb{N}$, the problem **p**-AT-MOST-r-EDGE-DELETION *admits:*

- *a constant factor approximation;*
- *a linear kernel.*

Application to immersion closed parameters

Let **p** be a parameter that is:

- closed under immersion and disjoint union;
- large on walls.

For instance: cutwidth, carving-width, tree-cut width, etc..

Theorem (Giannopoulou, Pilipczuk, Thilikos, R., Wrochna, 2016+)

For every $r \in \mathbb{N}$, the problem **p**-AT-MOST-r-EDGE-DELETION *admits:*

- *a constant factor approximation;*
- *a linear kernel.*

Mirrors the result of Fomin et al. on TREEWIDTH-AT-MOST-r-VERTEX-DELETION.

Recap

When graphs of $\mathcal{F}$ are connected and one of them is planar subcubic:

- constant factor approximation and linear kernel for $\mathcal{F}$-Immersion Deletion;
- linear bound on obstructions size for k-edge-distance to $\mathcal{F}$.

Recap

When graphs of $\mathcal{F}$ are connected and one of them is planar subcubic:

- constant factor approximation and linear kernel for $\mathcal{F}$-Immersion Deletion;
- linear bound on obstructions size for k-edge-distance to $\mathcal{F}$.

What do we learn?

- Immersion-Deletion $\neq$ Minor-Deletion
- tree-cut width might be the right parameter to study immersions.

Further work

1. lift the connectivity requirement;
2. consider the case where $\mathcal{F}$ contains does not contain a planar subcubic graph;
3. give a constructive bound on the number of obstructions.

Further work

1. lift the connectivity requirement;
2. consider the case where $\mathcal{F}$ contains does not contain a planar subcubic graph;
3. give a constructive bound on the number of obstructions.

Thank you for your attention!