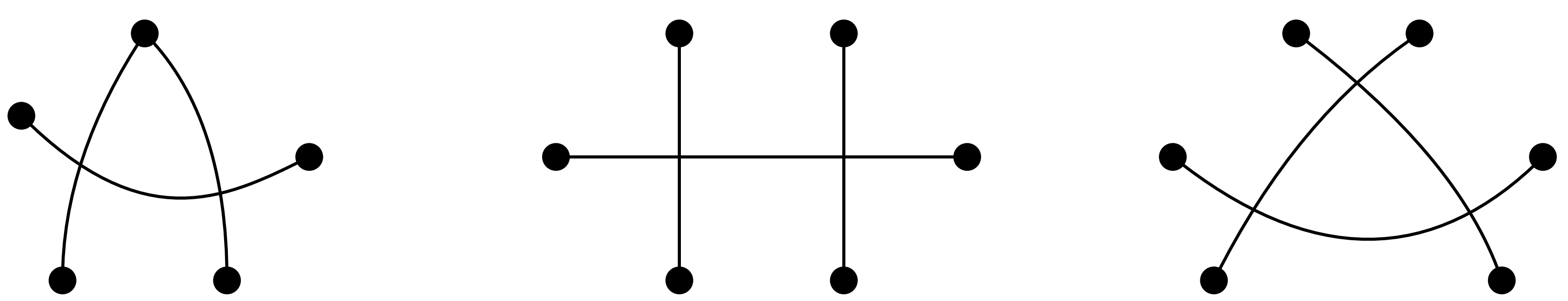
Universität Passau

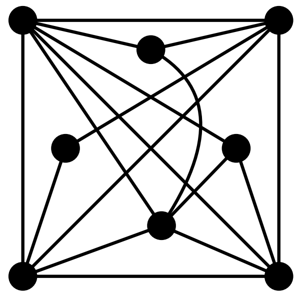
Parameterized Algorithms for Beyond Planar Crossing Numbers

Miriam Münch, Ignaz Rutter

18. September 2024

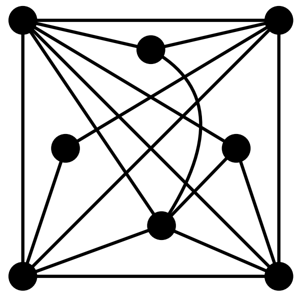


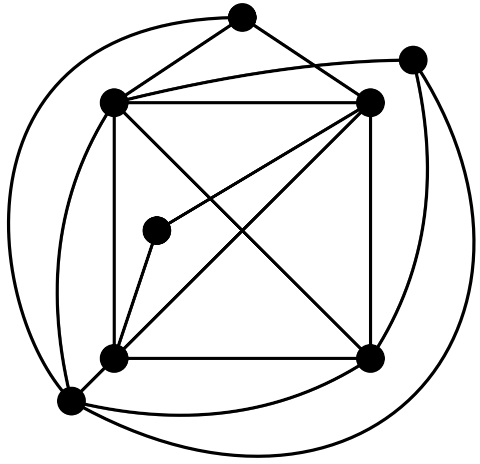
(Beyond-Planar) Crossing Numbers



crossing number of G = min. # crossings in any drawing of G

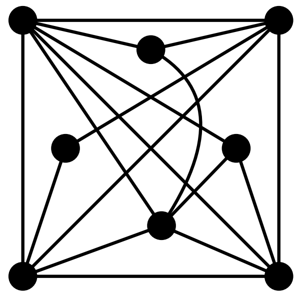
(Beyond-Planar) Crossing Numbers

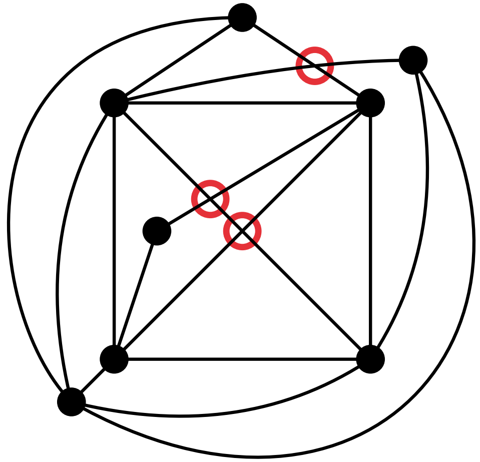




crossing number of G = min. # crossings in any drawing of G

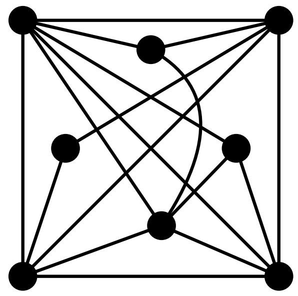
(Beyond-Planar) Crossing Numbers

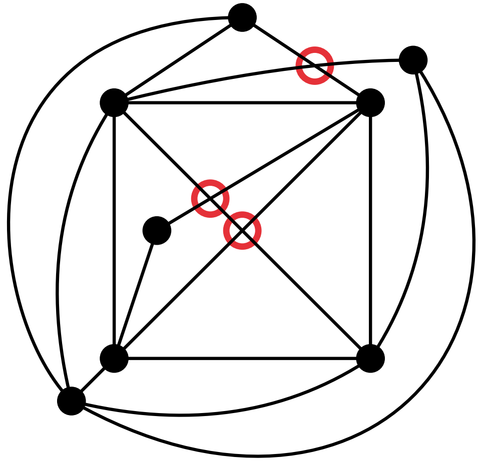




crossing number of G = min. # crossings in any drawing of G

(Beyond-Planar) Crossing Numbers

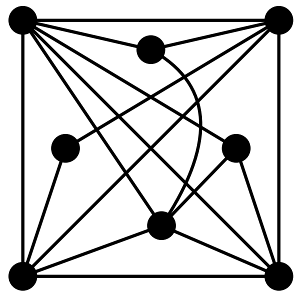


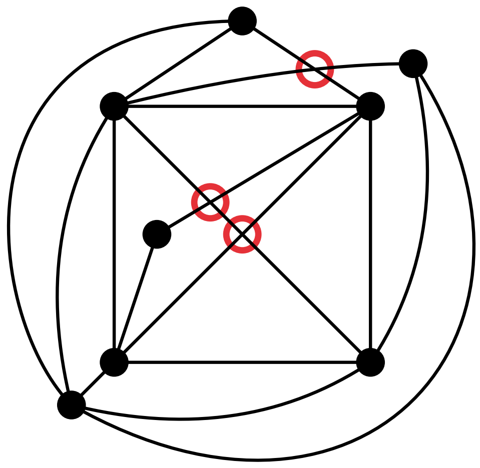


not 1-planar

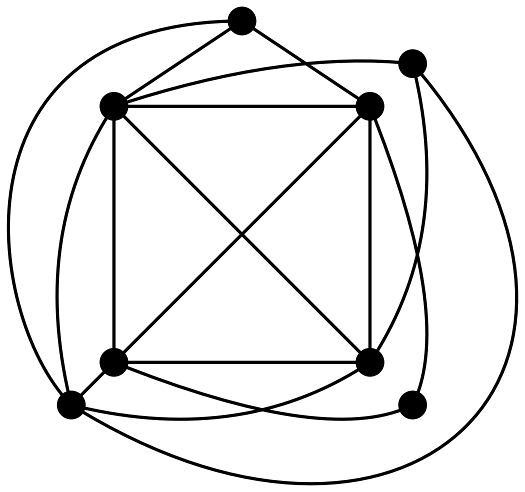
crossing number of *G* = min. # crossings in any   drawing of *G*

(Beyond-Planar) Crossing Numbers





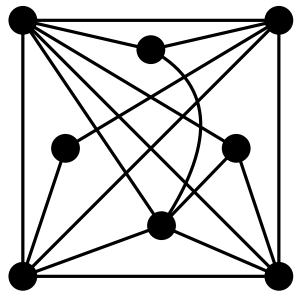
not 1-planar

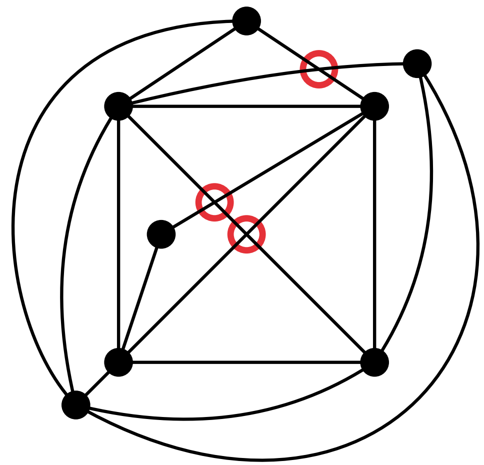


1-planar

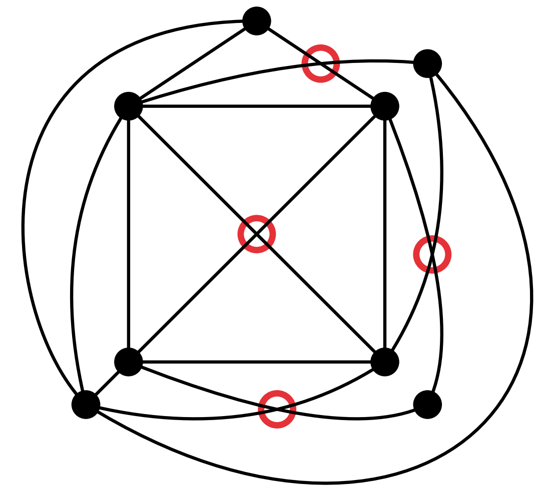
crossing number of G = min. # crossings in any drawing of G

(Beyond-Planar) Crossing Numbers





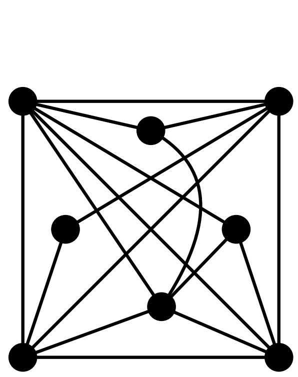
not 1-planar

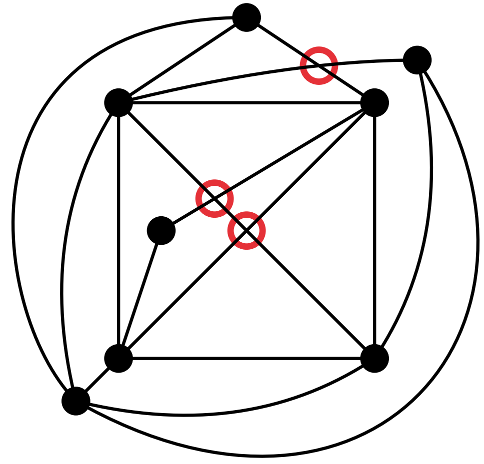


1-planar

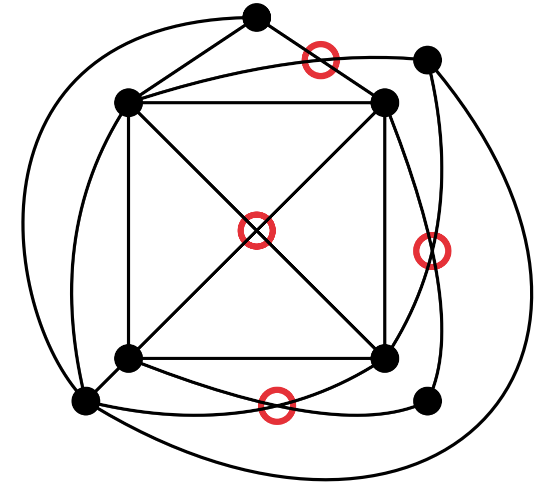
crossing number of *G* = min. # crossings in any drawing of *G*

(Beyond-Planar) Crossing Numbers





not 1-planar

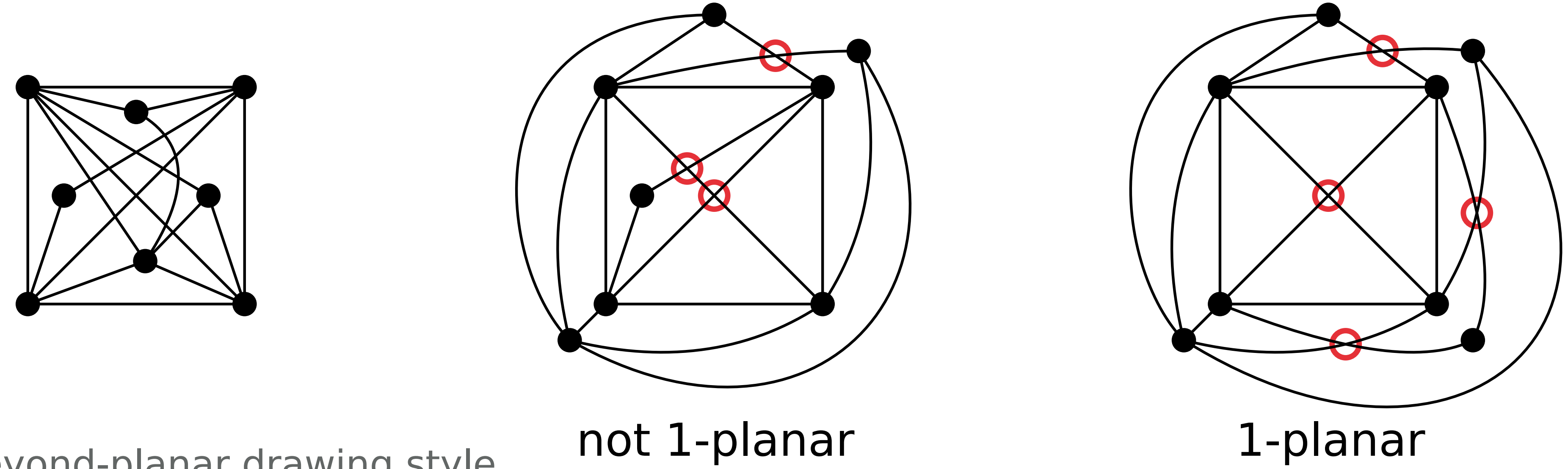


1-planar

beyond-planar drawing style

$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

UNIVERSITÄT PASSAU



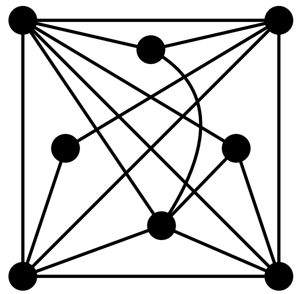
beyond-planar drawing style

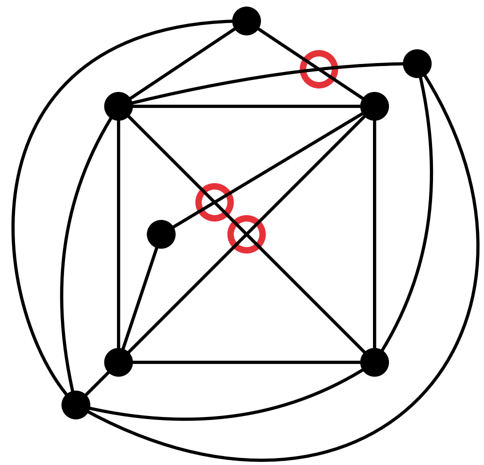
$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

- can deviate significantly from each other [Chimani et al., Beusekom et al.]

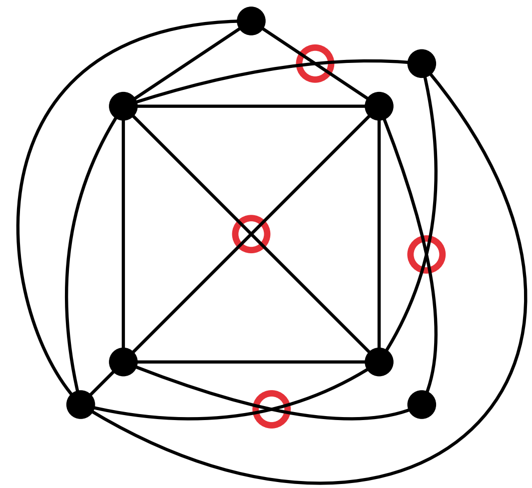
(Beyond-Planar) Crossing Numbers

UNIVERSITÄT PASSAU





not 1-planar



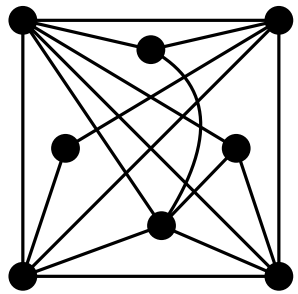
1-planar

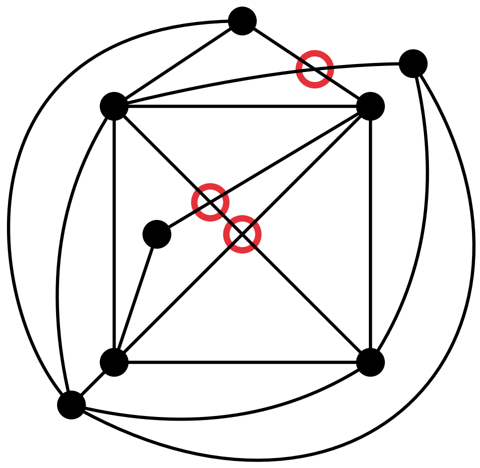
beyond-planar drawing style

$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

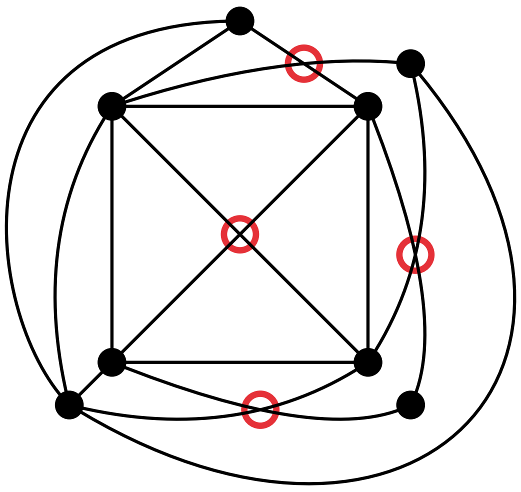
- can deviate significantly from each other [Chimani et al., Beusekom et al.]
- computing crossing number is NP-complete [Garey, Johnson]

(Beyond-Planar) Crossing Numbers





not 1-planar



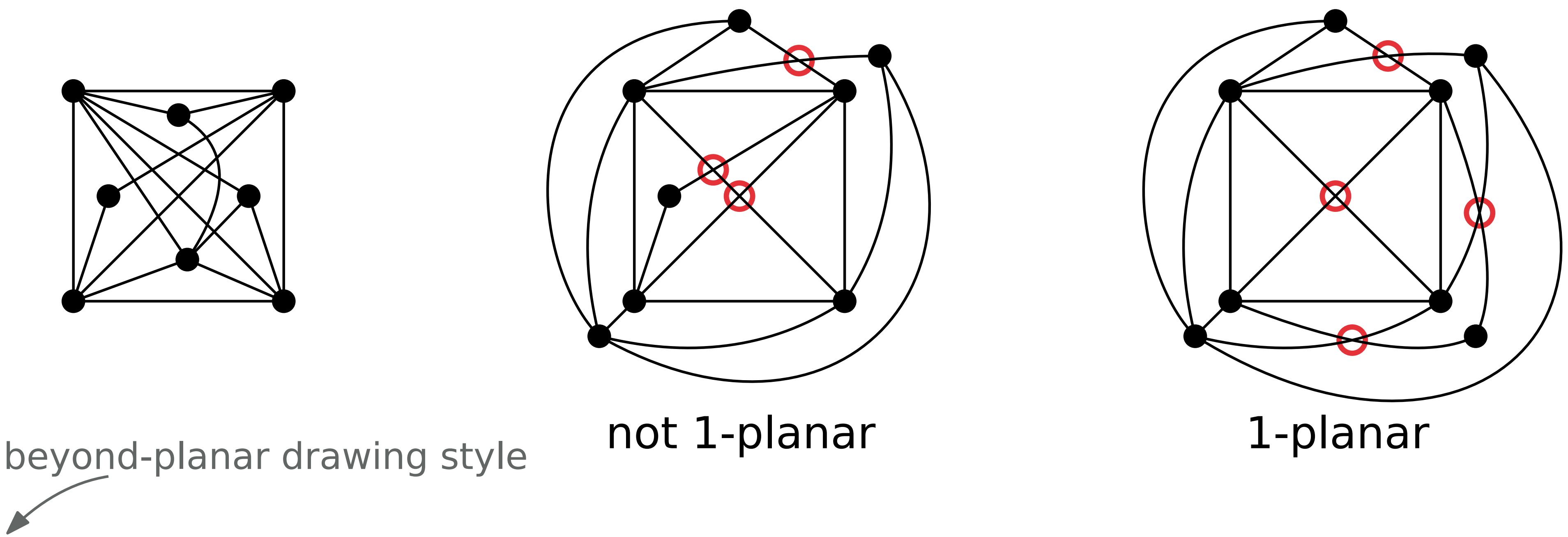
1-planar

beyond-planar drawing style

$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

- can deviate significantly from each other [Chimani et al., Beusekom et al.]
- computing crossing number is NP-complete [Garey, Johnson]
- crossing number can be computed in FPT-time [Grohe]

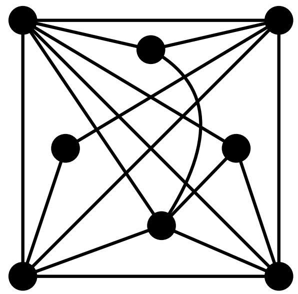
(Beyond-Planar) Crossing Numbers

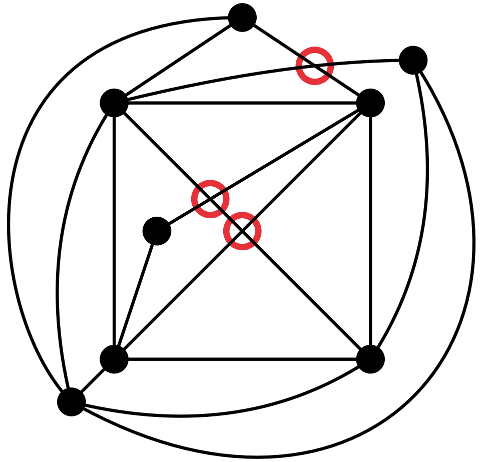


$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

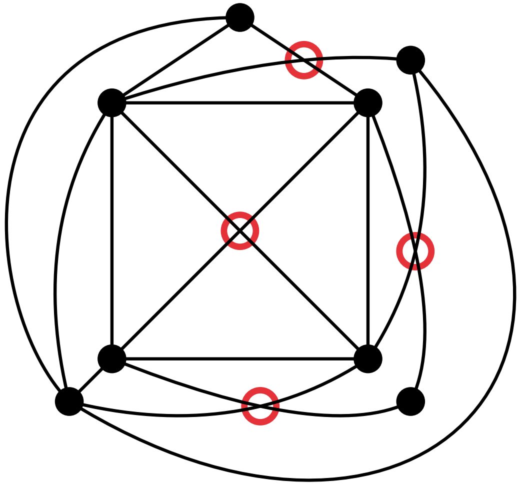
- can deviate significantly from each other [Chimani et al., Beusekom et al.]
- computing crossing number is NP-complete [Garey, Johnson]
- crossing number can be computed in FPT-time [Grohe]

Can decide whether $\text{crn}(G) \leq c$ in time $f(c)n^{O(1)}$





not 1-planar



1-planar

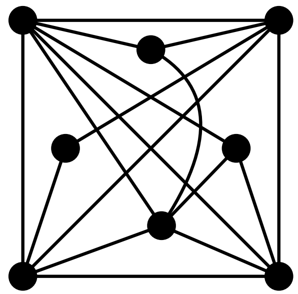
beyond-planar drawing style

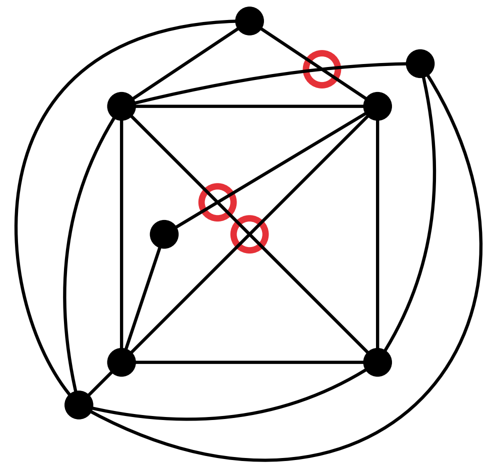
$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

- can deviate significantly from each other [Chimani et al., Beusekom et al.]
- recognizing beyond-planar graph classes often NP-complete

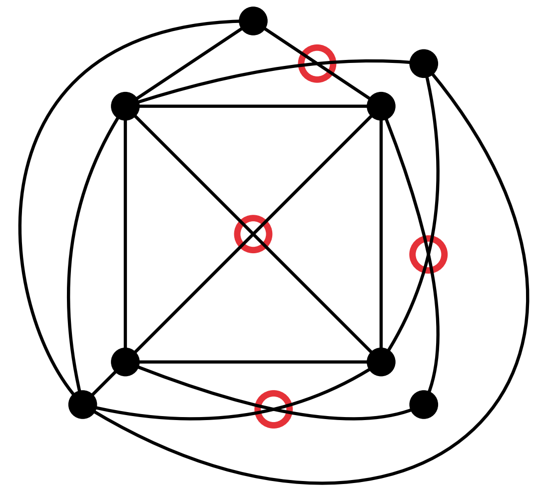
Can decide whether $\text{crn}(G) \leq c$ in time $f(c)n^{O(1)}$

(Beyond-Planar) Crossing Numbers





not 1-planar



1-planar

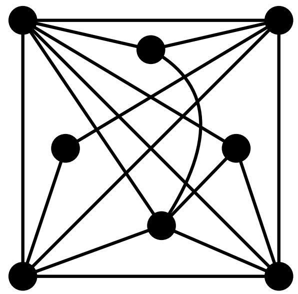
beyond-planar drawing style

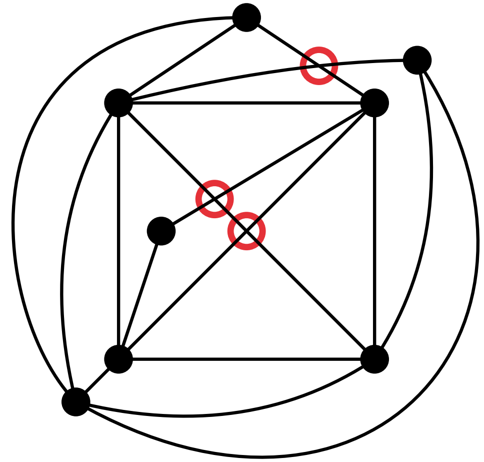
$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

- can deviate significantly from each other [Chimani et al., Beusekom et al.]
- recognizing beyond-planar graph classes often NP-complete
- 1-planar crossing number can be computed in FPT-time [Hamm, Hliněný]

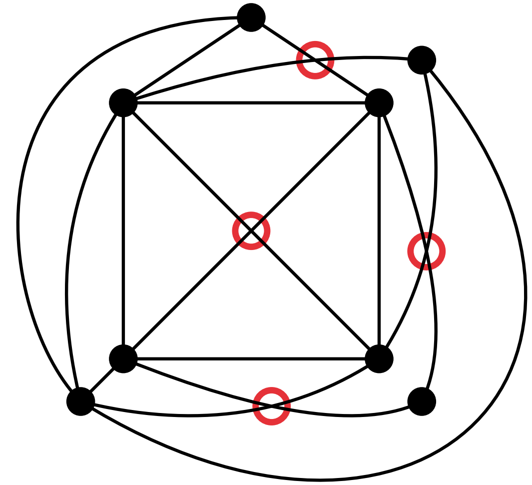
Can decide whether $\text{crn}(G) \leq c$ in time $f(c)n^{O(1)}$

(Beyond-Planar) Crossing Numbers





not 1-planar



1-planar

beyond-planar drawing style

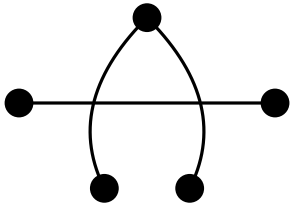
$\mathcal{D}$- crossing number of G = min. # crossings in any $\mathcal{D}$-drawing of G

- can deviate significantly from each other [Chimani et al., Beusekom et al.]
- recognizing beyond-planar graph classes often NP-complete
- 1-planar crossing number can be computed in FPT-time [Hamm, Hliněný]

Can we decide $\mathcal{D}$-crn$(G) \leq c$ in time $f(c)n^{O(1)}$?

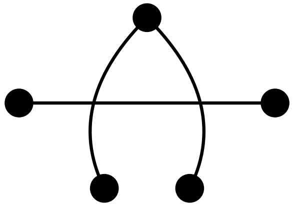
Main Result

Beyond-planar graph classes usually defined via forbidden patterns



Main Result

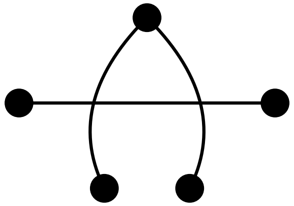
Beyond-planar graph classes usually defined via forbidden patterns



Goal: Combinatorial formalization

Main Result

Beyond-planar graph classes usually defined via forbidden patterns

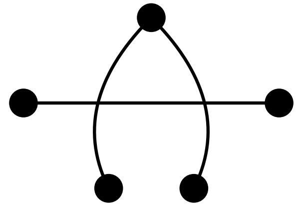


Goal: Combinatorial formalization

Meta-Theorem

For any set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$ is FPT w.r.t. c.

Beyond-planar graph classes usually defined via forbidden patterns



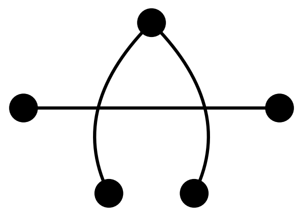
Goal: Combinatorial formalization

Meta-Theorem

For any set $\mathcal{F}$ of crossing patterns, testing whether a graph *G* admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$ is FPT w.r.t. *c*.

For any beyond-planar graph class $\mathcal{D}$ that is defined by finitely many forbidden patterns, computing $\mathcal{D}$-crn(G) is FPT.

Beyond-planar graph classes usually defined via forbidden patterns



Goal: Combinatorial formalization

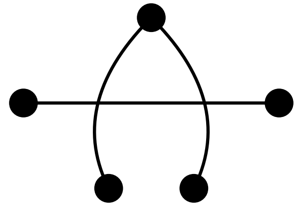
Meta-Theorem

For any set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$ is FPT w.r.t. c.

For any beyond-planar graph class $\mathcal{D}$ that is defined by finitely many forbidden patterns, computing $\mathcal{D}$-crn(G) is FPT.

based on Grohe's approach

Beyond-planar graph classes usually defined via forbidden patterns



Goal: Combinatorial formalization

Meta-Theorem

For any set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$ is FPT w.r.t. c.

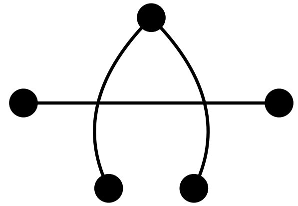
For any beyond-planar graph class $\mathcal{D}$ that is defined by finitely many forbidden patterns, computing $\mathcal{D}$-crn(G) is FPT.

based on Grohe's approach

Phase 1: Bound treewidth w.r.t. beyond-planar crossing number

Main Result

Beyond-planar graph classes usually defined via forbidden patterns



Goal: Combinatorial formalization

Meta-Theorem

For any set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$ is FPT w.r.t. c.

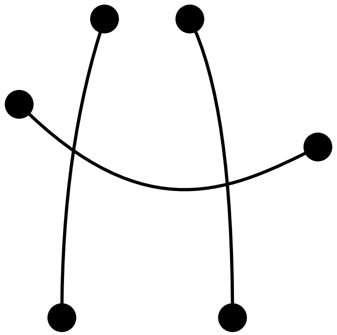
For any beyond-planar graph class $\mathcal{D}$ that is defined by finitely many forbidden patterns, computing $\mathcal{D}$-crn(G) is FPT.

based on Grohe's approach

Phase 1: Bound treewidth w.r.t. beyond-planar crossing number

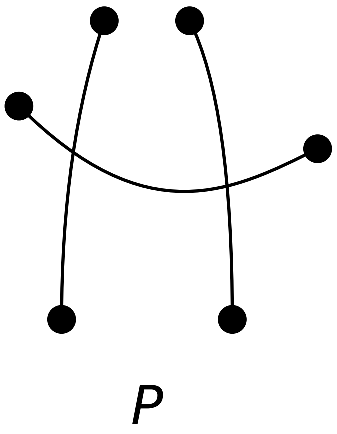
Phase 2: Solve problem on graphs of bounded treewidth via Courcelle

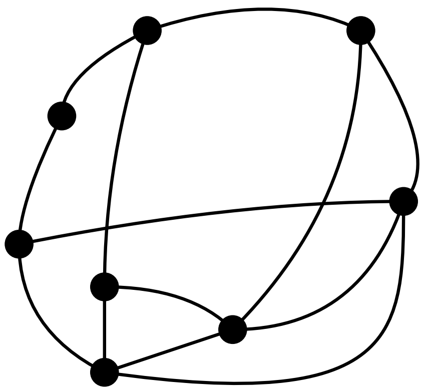
Crossing Patterns



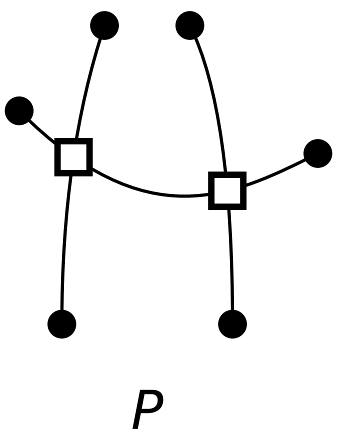
P

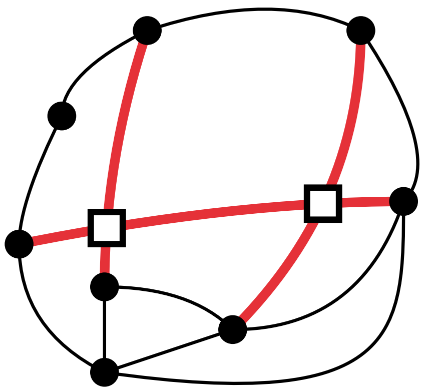
Crossing Patterns



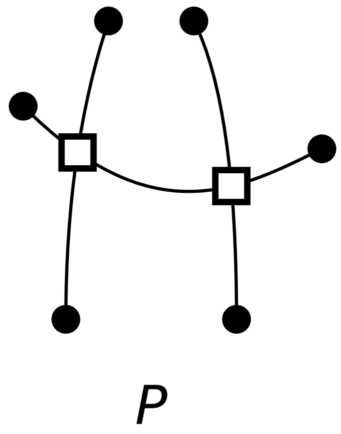


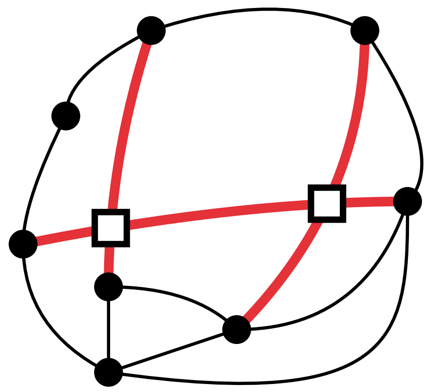
Crossing Patterns





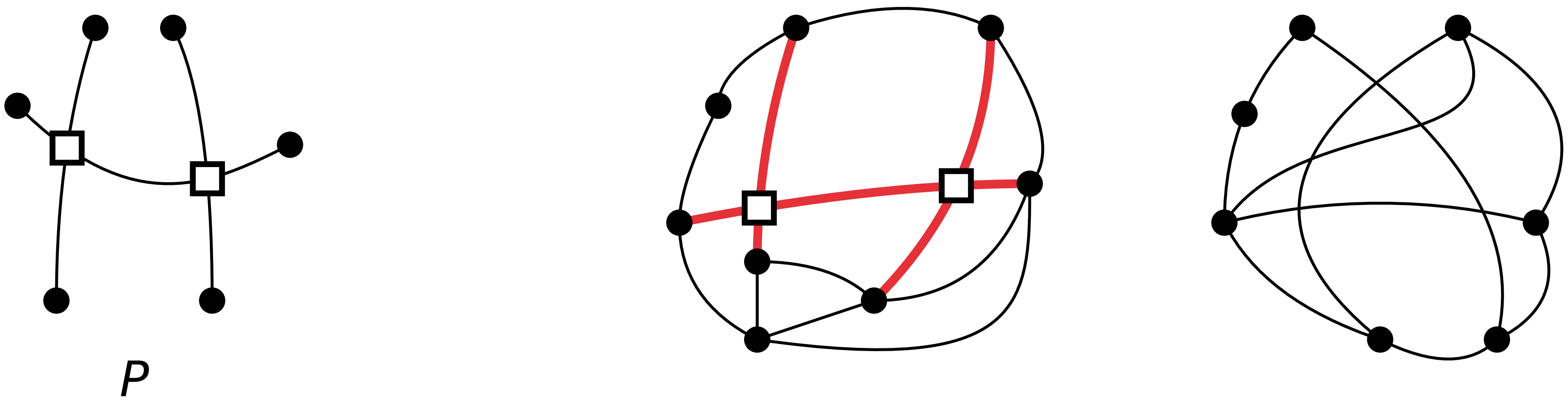
Crossing Patterns





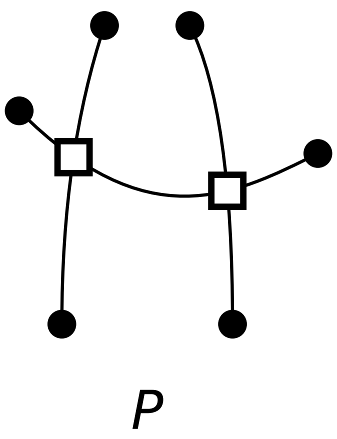
- crossing in $P \rightarrow$ crossing in drawing required

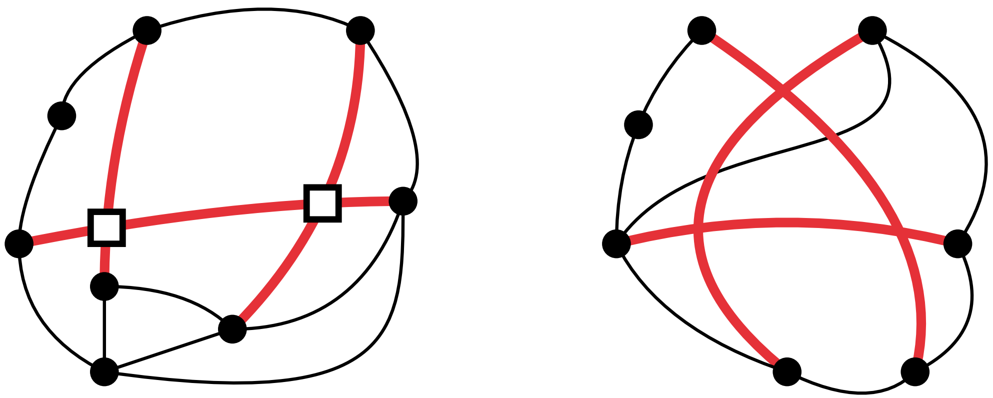
Crossing Patterns



- crossing in $P \rightarrow$ crossing in drawing required

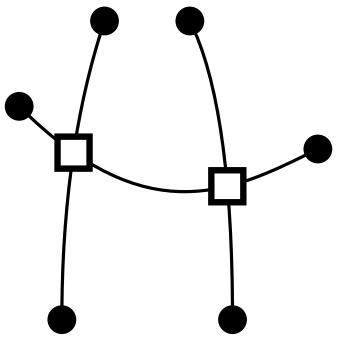
Crossing Patterns



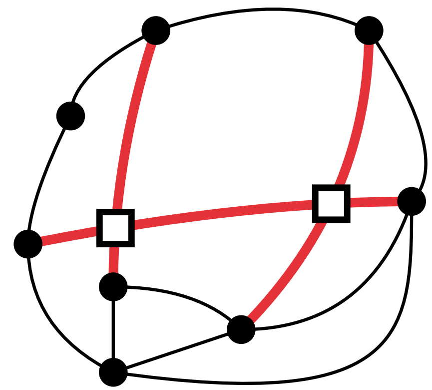


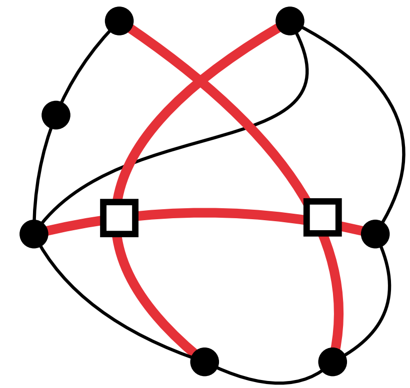
- crossing in $P \rightarrow$ crossing in drawing required

Crossing Patterns



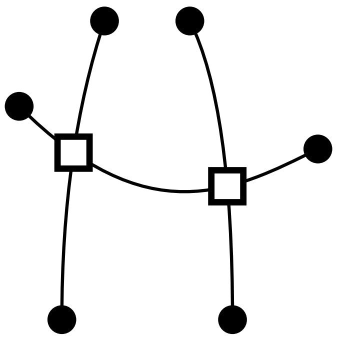
P



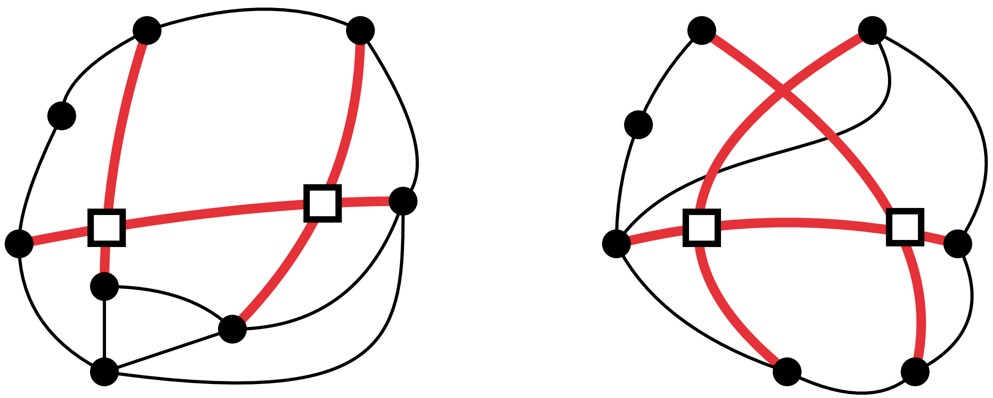


- crossing in *P* $\rightarrow$ crossing in drawing required

Crossing Patterns

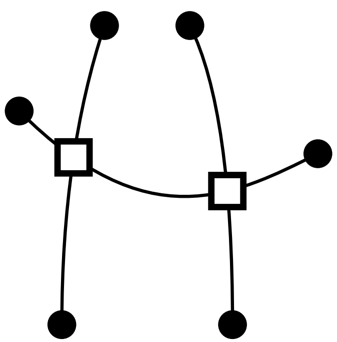


P

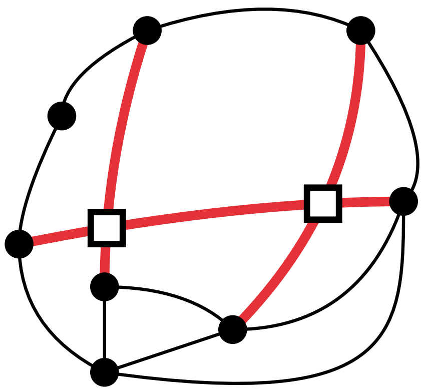


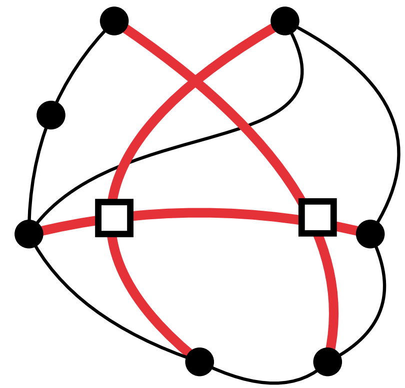
- crossing in *P* $\rightarrow$ crossing in drawing required
- crossing in drawing that is absent in *P* does not help to avoid *P*

Crossing Patterns



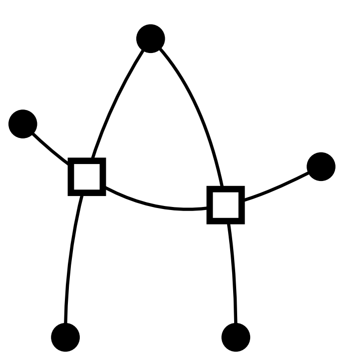
P

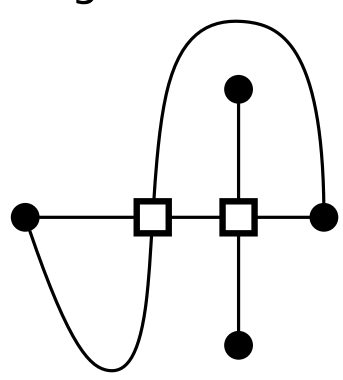


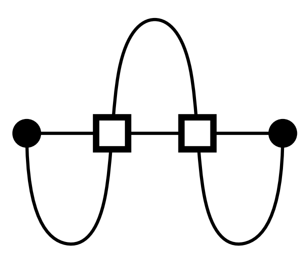


- crossing in *P* $\rightarrow$ crossing in drawing required
- crossing in drawing that is absent in *P* does not help to avoid *P*

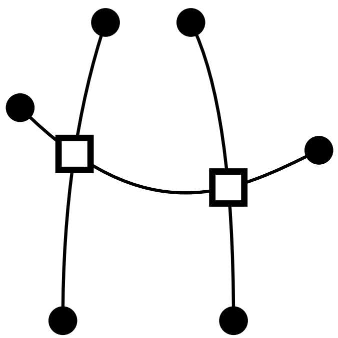
Forbidden in 1-planar drawings:



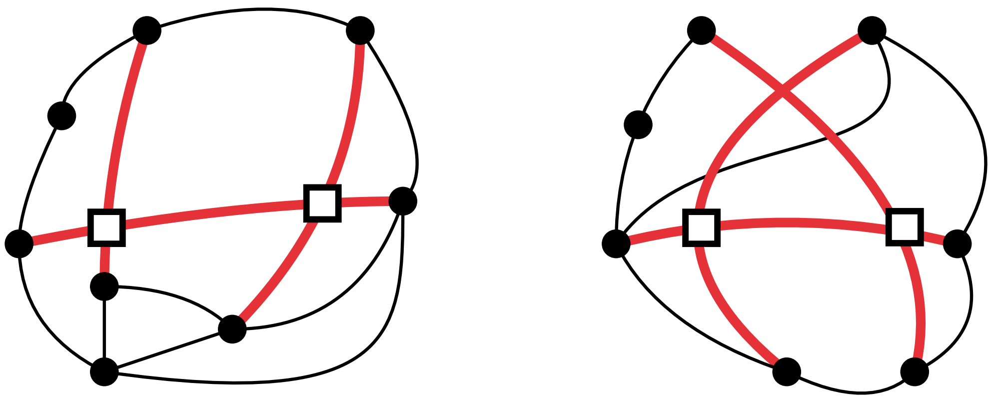




Crossing Patterns

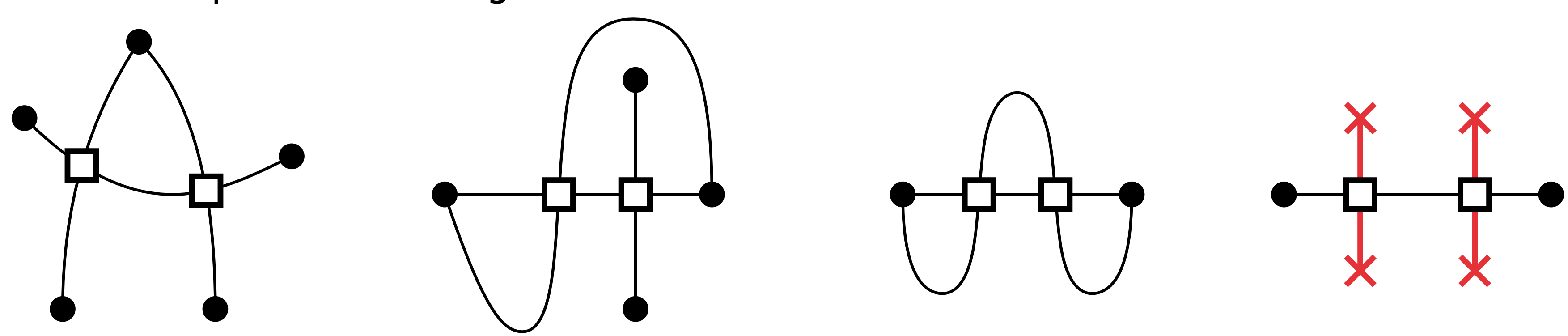


P

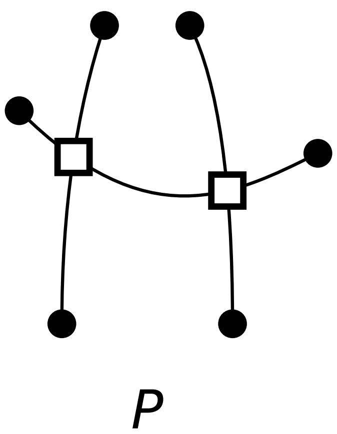


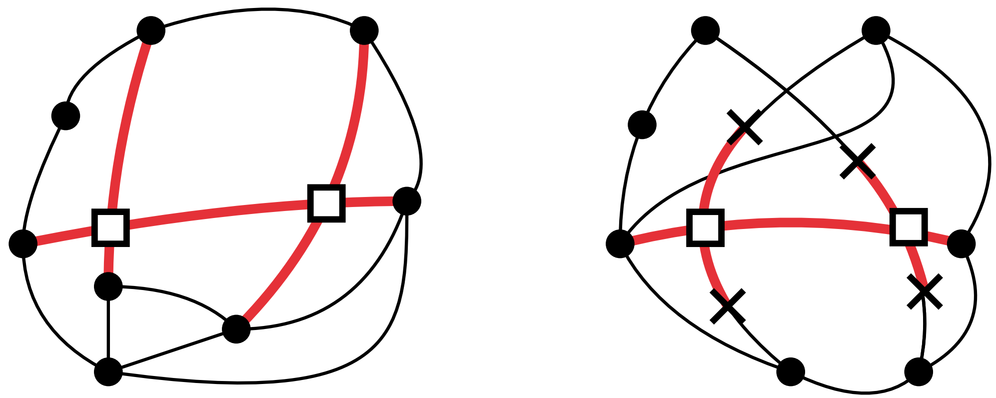
- crossing in $P \rightarrow$ crossing in drawing required
- crossing in drawing that is absent in *P* does not help to avoid *P*
- allow mapping edge in *P* to part of edge in drawing

Forbidden in 1-planar drawings:



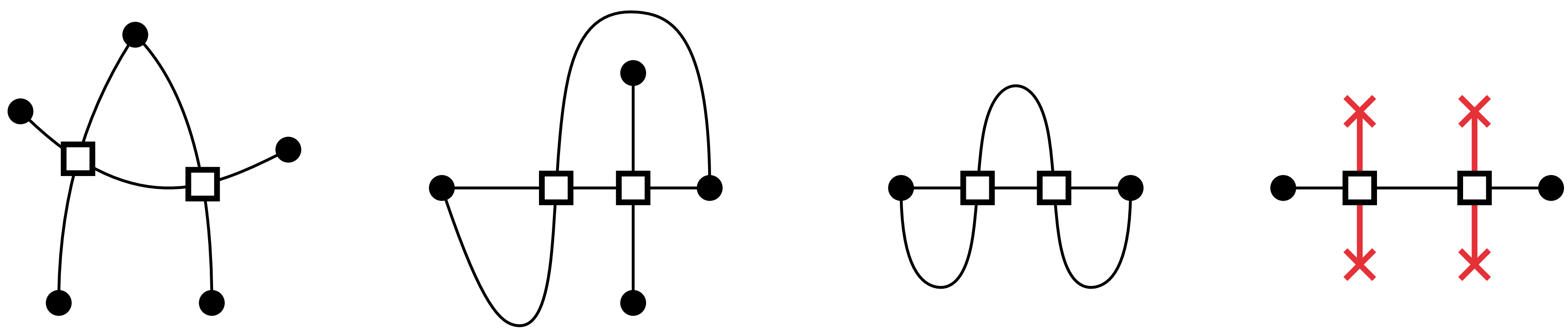
Crossing Patterns





- crossing in $P \rightarrow$ crossing in drawing required
- crossing in drawing that is absent in P does not help to avoid P
- allow mapping edge in P to part of edge in drawing

Forbidden in 1-planar drawings:

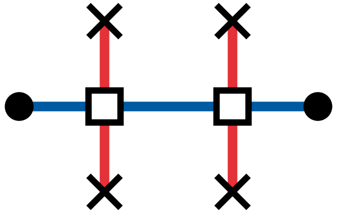


Crossing Patterns – Definition

real / crossing / subdivision

A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

Forbidden Patterns



1-planar

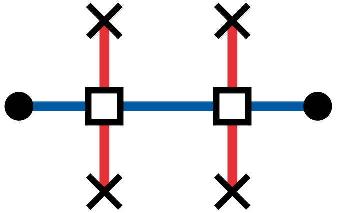
Crossing Patterns – Definition

real / crossing / subdivision

A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

- each vertex in S has degree 1,

Forbidden Patterns



1-planar

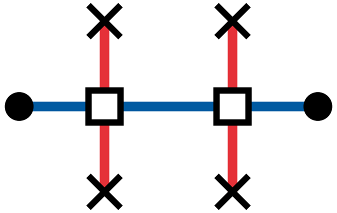
Crossing Patterns – Definition

real / crossing / subdivision

A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

- each vertex in S has degree 1,
- each vertex in C has degree 4,

Forbidden Patterns



1-planar

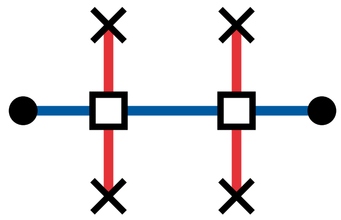
Crossing Patterns – Definition

real / crossing / subdivision

A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

- each vertex in S has degree 1,
- each vertex in C has degree 4,
- each vertex in R and each vertex in S has at least one neighbor in C.

Forbidden Patterns



1-planar

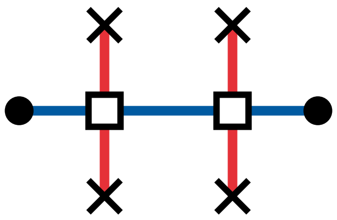
Crossing Patterns – Definition

real / crossing / subdivision

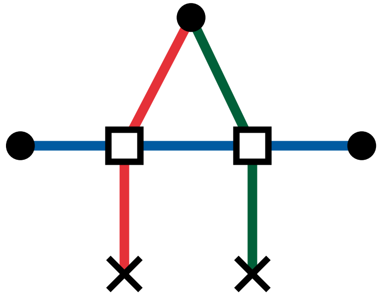
A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

- each vertex in S has degree 1,
- each vertex in C has degree 4,
- each vertex in R and each vertex in S has at least one neighbor in C.

Forbidden Patterns



1-planar



fan-crossing free

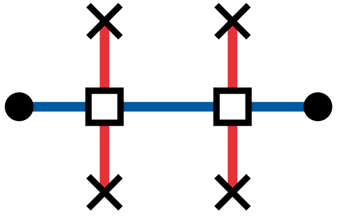
Crossing Patterns – Definition

real / crossing / subdivision

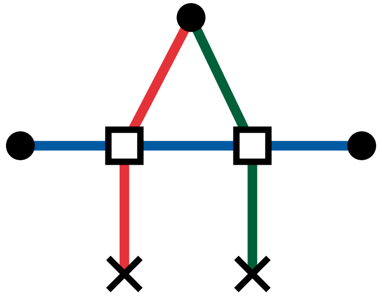
A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

- each vertex in S has degree 1,
- each vertex in C has degree 4,
- each vertex in R and each vertex in S has at least one neighbor in C.

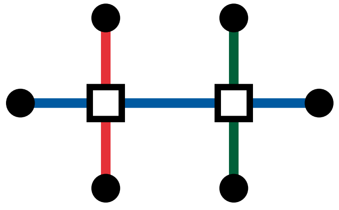
Forbidden Patterns



1-planar



fan-crossing free



fan-crossing

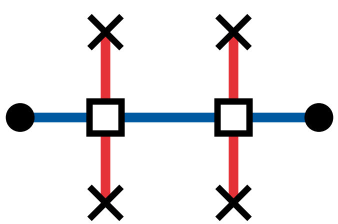
Crossing Patterns – Definition

real / crossing / subdivision

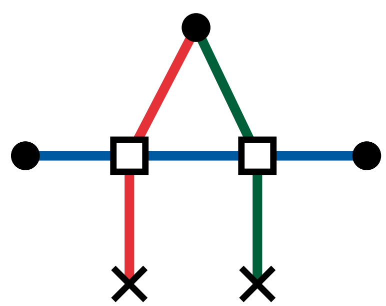
A **crossing pattern** is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ s.t.

- each vertex in S has degree 1,
- each vertex in C has degree 4,
- each vertex in R and each vertex in S has at least one neighbor in C.

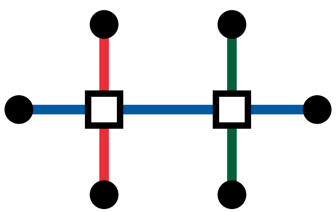
Forbidden Patterns



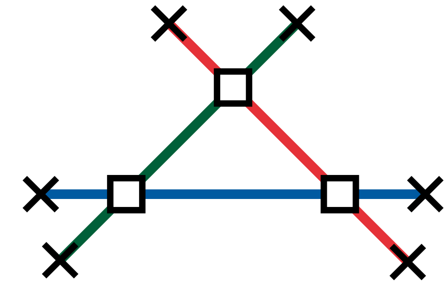
1-planar



fan-crossing free



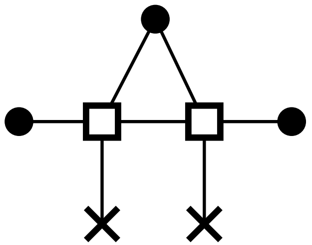
fan-crossing

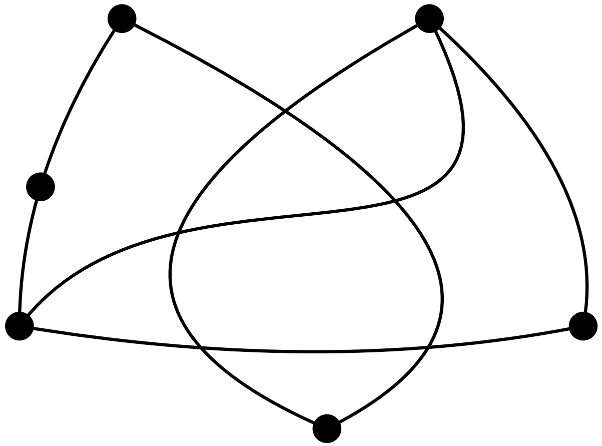


quasi-planar

Containing a Pattern

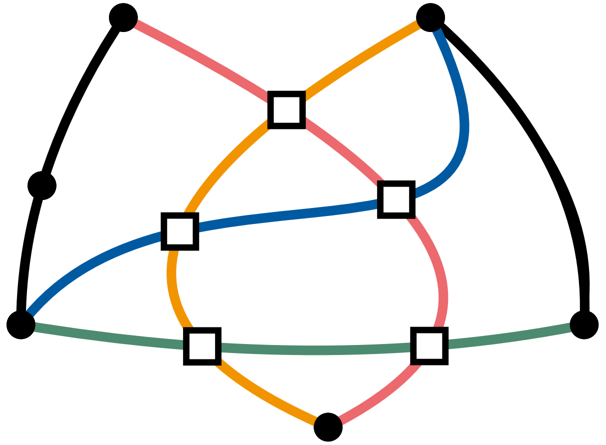
Fan-crossing free?

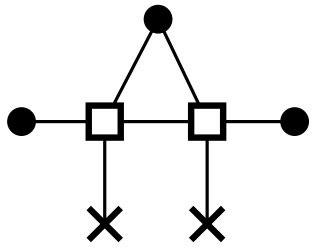




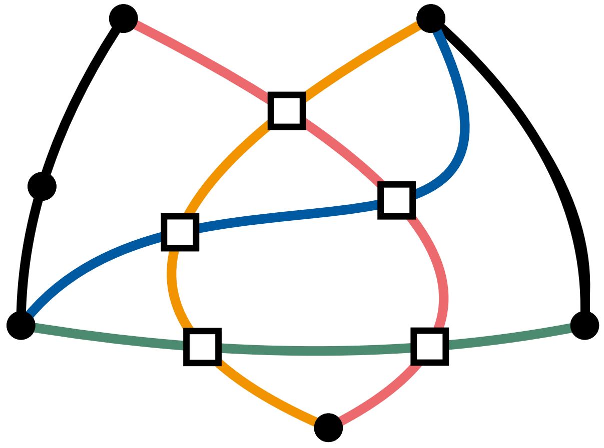
Containing a Pattern

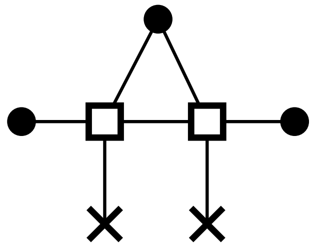
Fan-crossing free?





Fan-crossing free?

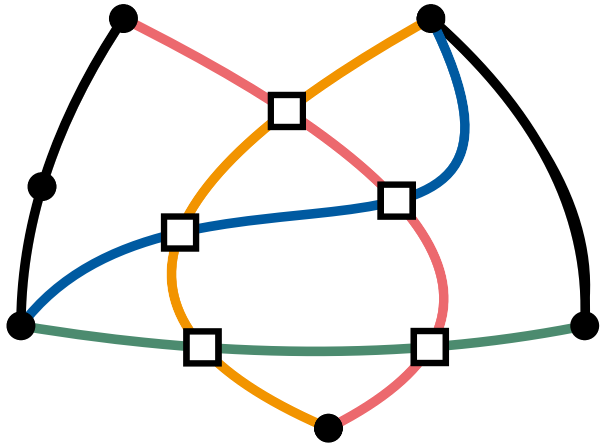


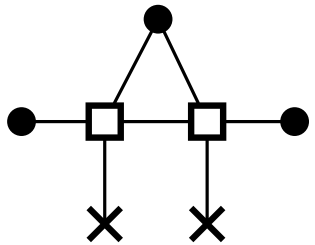


Allowed Operations:

(*i*) delete isolated vertices

Fan-crossing free?





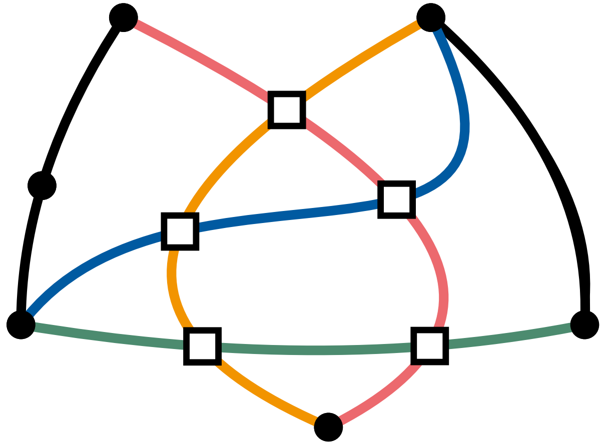
Allowed Operations:

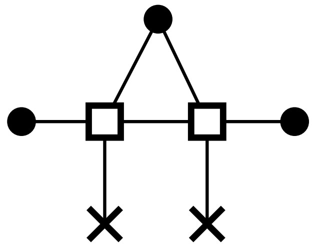
(*i*) delete isolated vertices

(*ii*) subdivide edge by introducing subdivision vertex

Containing a Pattern

Fan-crossing free?





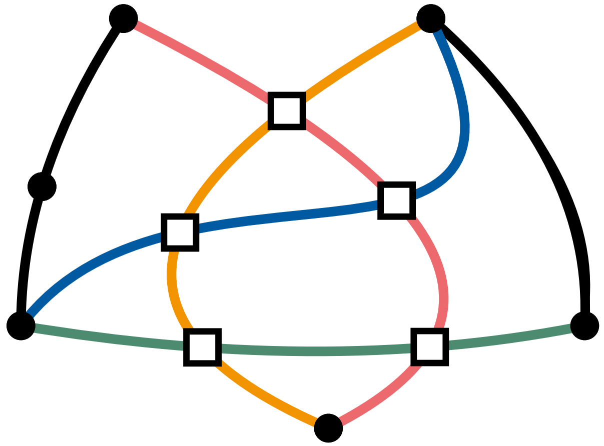
Allowed Operations:

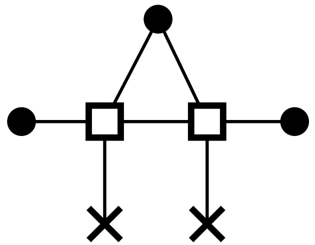
(*i*) delete isolated vertices

(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

Fan-crossing free?



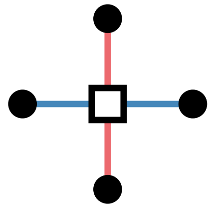


Allowed Operations:

(*i*) delete isolated vertices

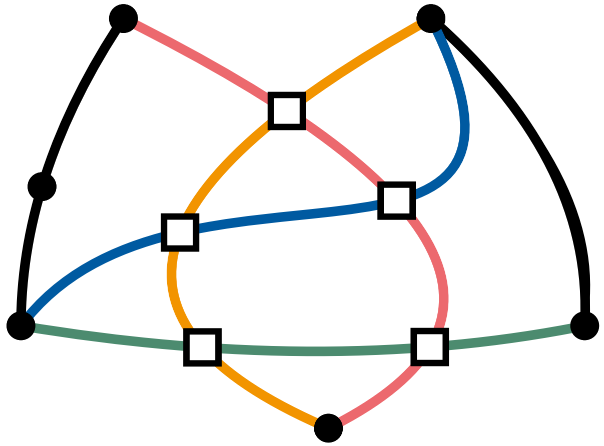
(*ii*) subdivide edge by introducing subdivision vertex

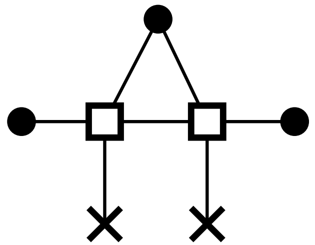
(*iii*) smooth a crossing



Containing a Pattern

Fan-crossing free?



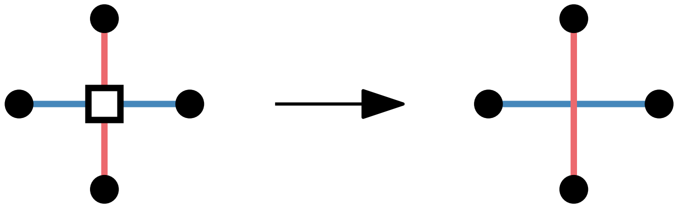


Allowed Operations:

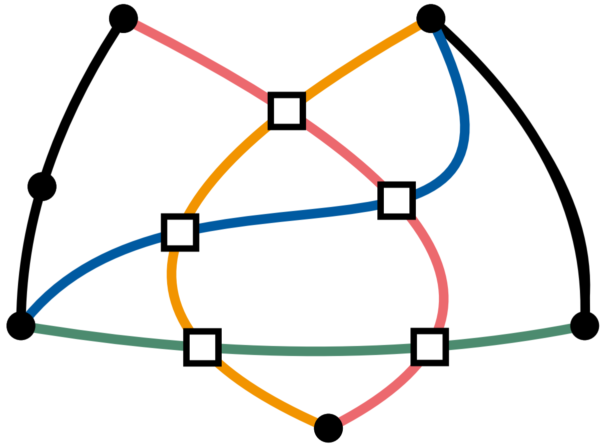
(*i*) delete isolated vertices

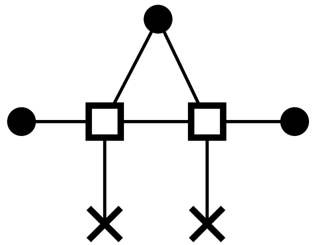
(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing



Fan-crossing free?





Allowed Operations:

(*i*) delete isolated vertices

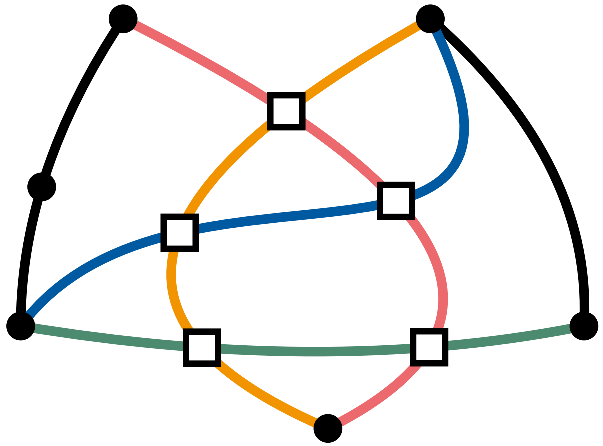
(*ii*) subdivide edge by introducing subdivision vertex

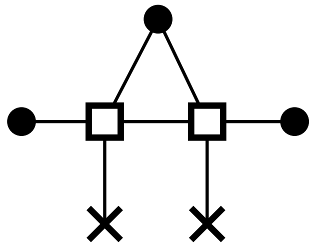
(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Containing a Pattern

Fan-crossing free?





Allowed Operations:

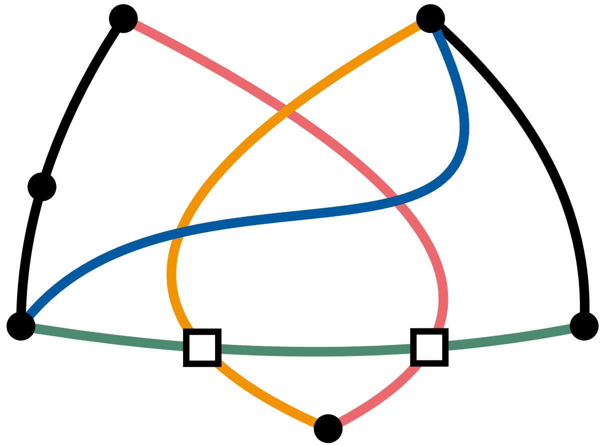
(*i*) delete isolated vertices

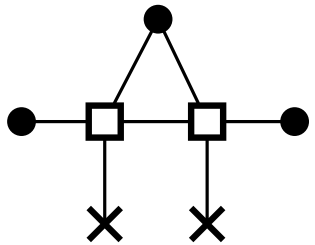
(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Fan-crossing free?





Allowed Operations:

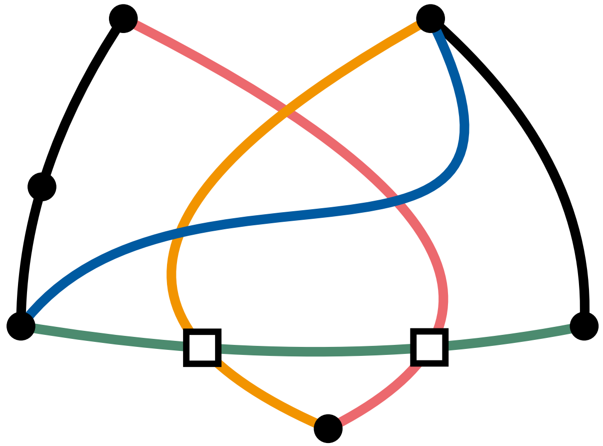
(*i*) delete isolated vertices

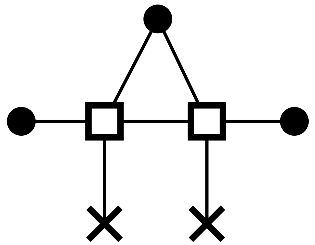
(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Fan-crossing free?





Allowed Operations:

(*i*) delete isolated vertices

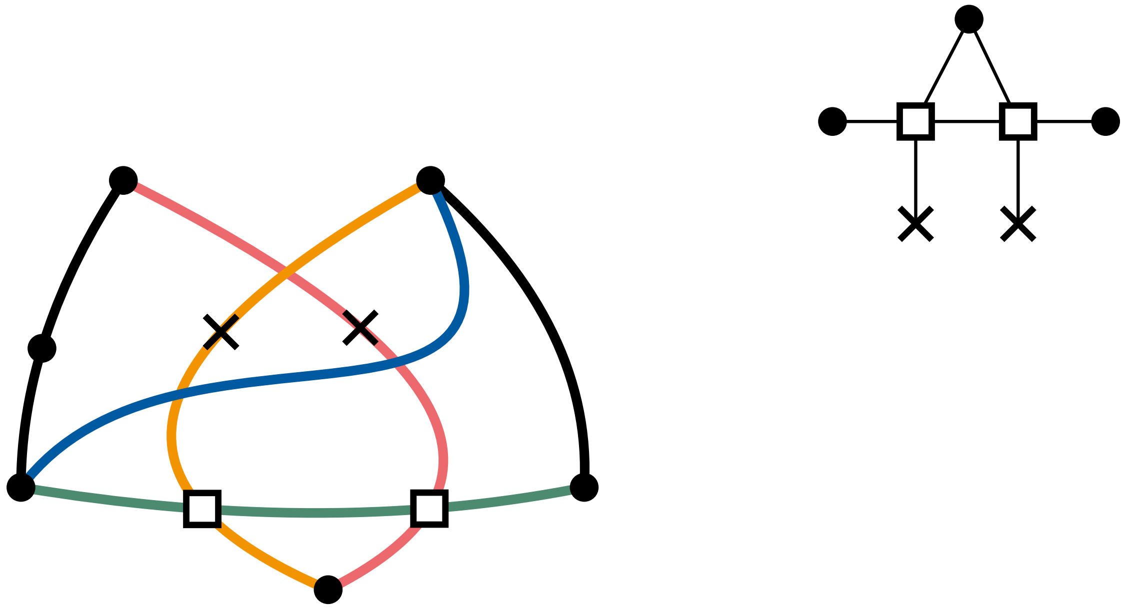
(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Containing a Pattern

Fan-crossing free?



Allowed Operations:

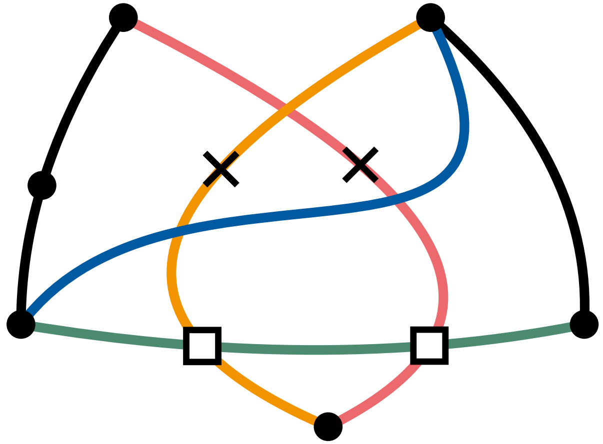
(*i*) delete isolated vertices

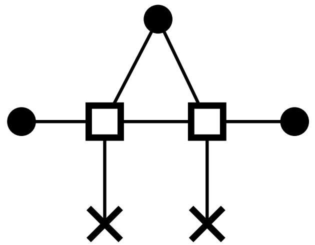
(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Fan-crossing free?





Allowed Operations:

(*i*) delete isolated vertices

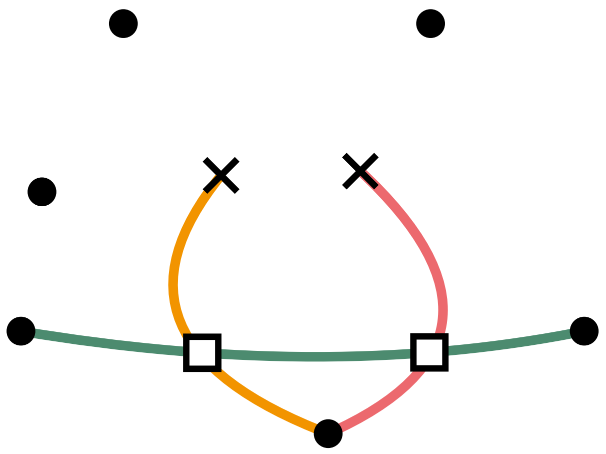
(*ii*) subdivide edge by introducing subdivision vertex

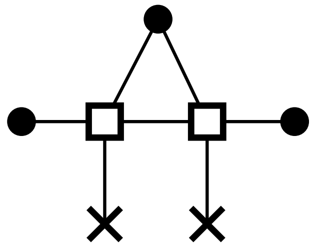
(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Containing a Pattern

Fan-crossing free?





Allowed Operations:

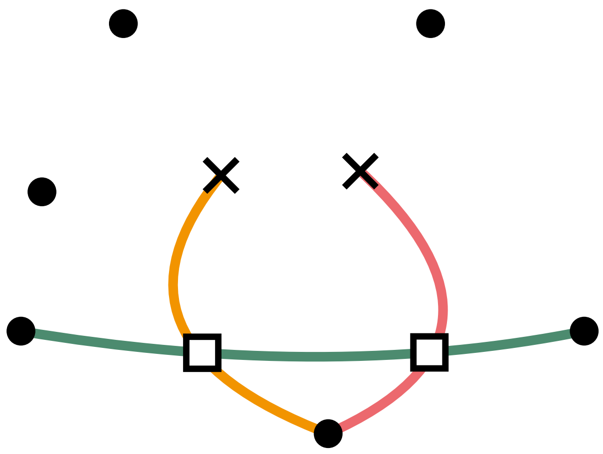
(*i*) delete isolated vertices

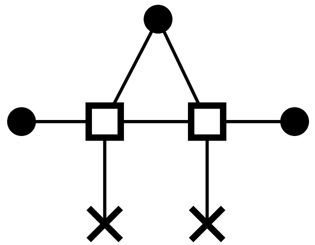
(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Fan-crossing free?





Allowed Operations:

(*i*) delete isolated vertices

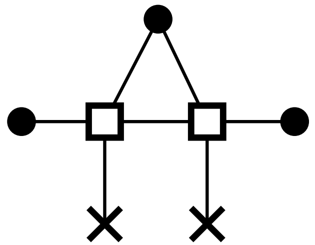
(*ii*) subdivide edge by introducing subdivision vertex

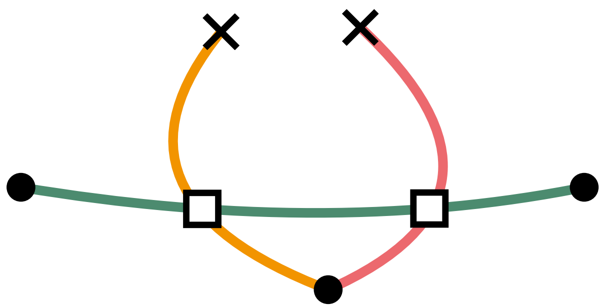
(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Containing a Pattern

Fan-crossing free?





Allowed Operations:

(*i*) delete isolated vertices

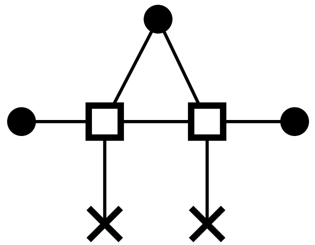
(*ii*) subdivide edge by introducing subdivision vertex

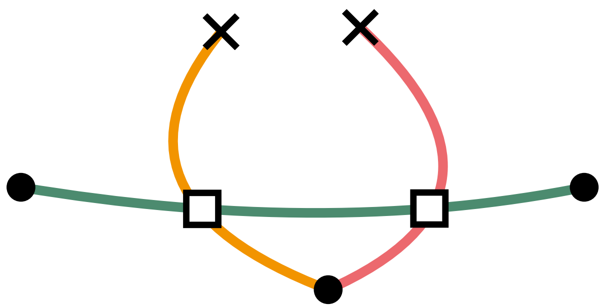
(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Containing a Pattern

Fan-crossing free? No!





Allowed Operations:

(*i*) delete isolated vertices

(*ii*) subdivide edge by introducing subdivision vertex

(*iii*) smooth a crossing

(*iv*) delete edge that is not incident to a crossing vertex

Bounding Treewidth

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Bounding Treewidth

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings,

Bounding Treewidth

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings, reject!

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings, reject!
- recognizes $\text{tw}(G) \leq w$ or

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings, reject!
- recognizes $\mathrm{tw}(G) \leq w$ or move to Phase 2

Bounding Treewidth

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings, reject!
- recognizes $\text{tw}(G) \leq w$ or move to Phase 2
- finds large hex-grid in G.

Goal:

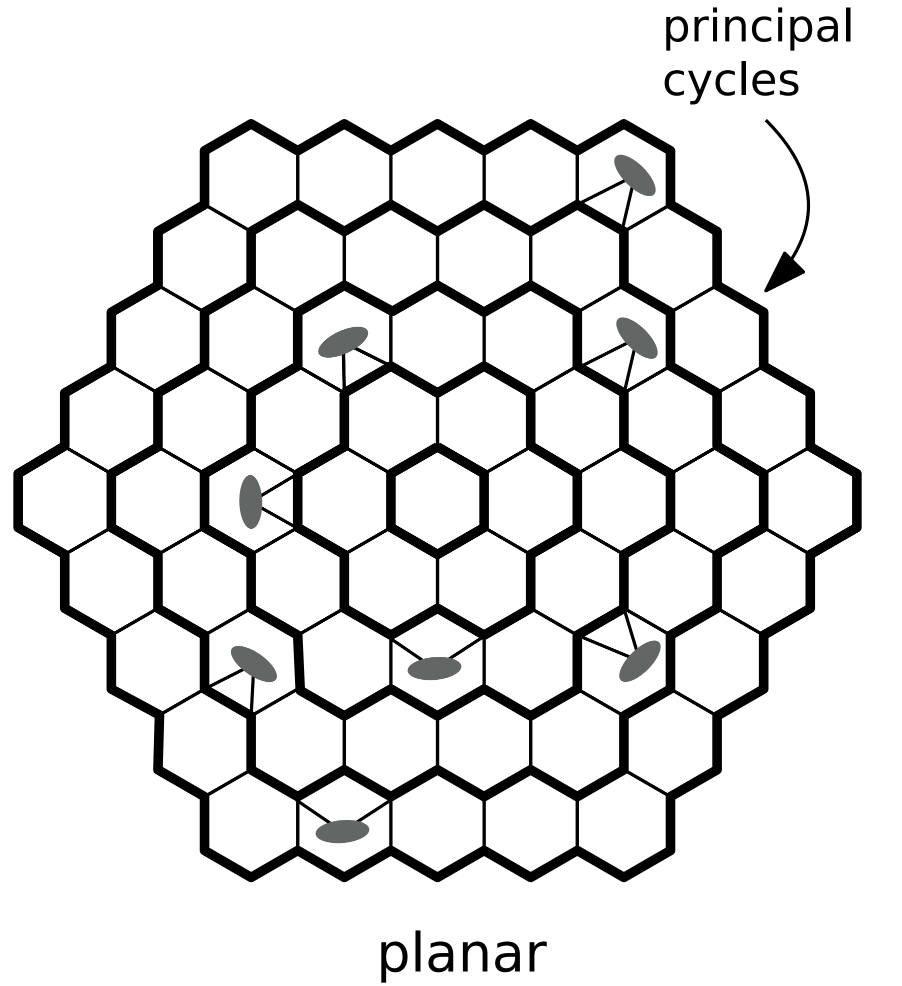
For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

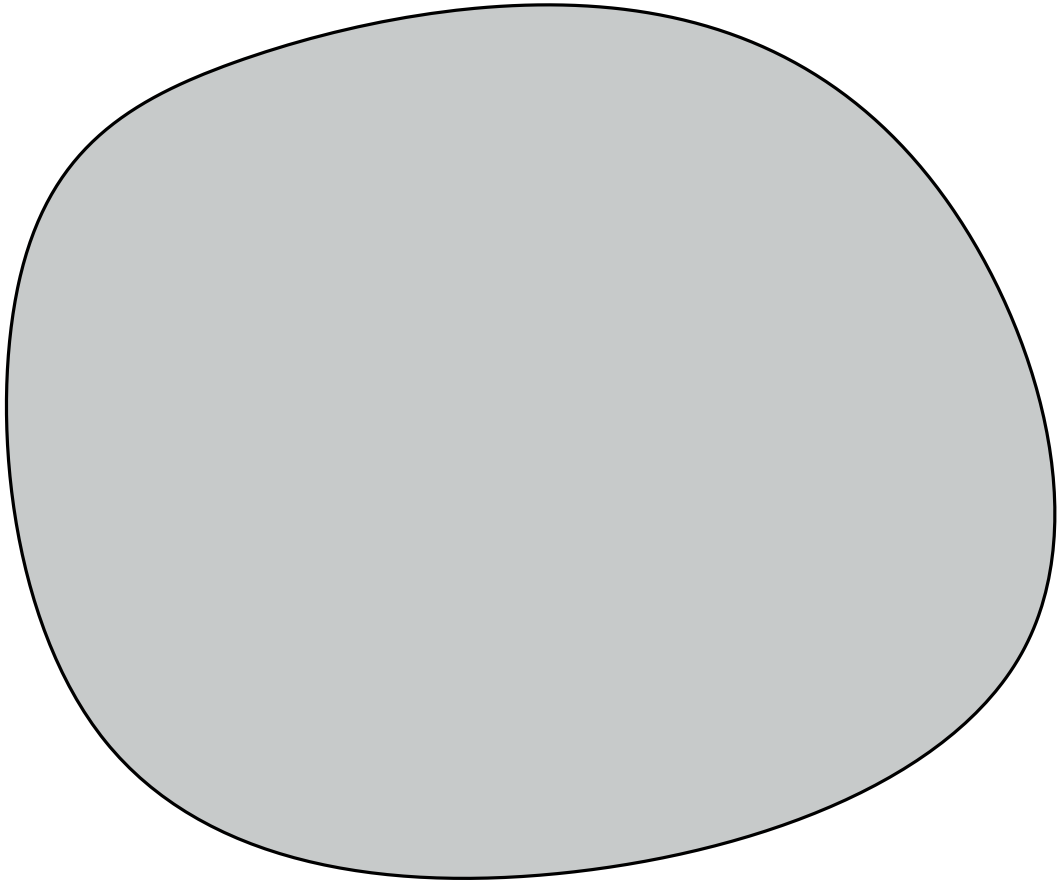
There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings, reject!
- recognizes $\text{tw}(G) \leq w$ or move to Phase 2
- finds large hex-grid in G. **Todo!**

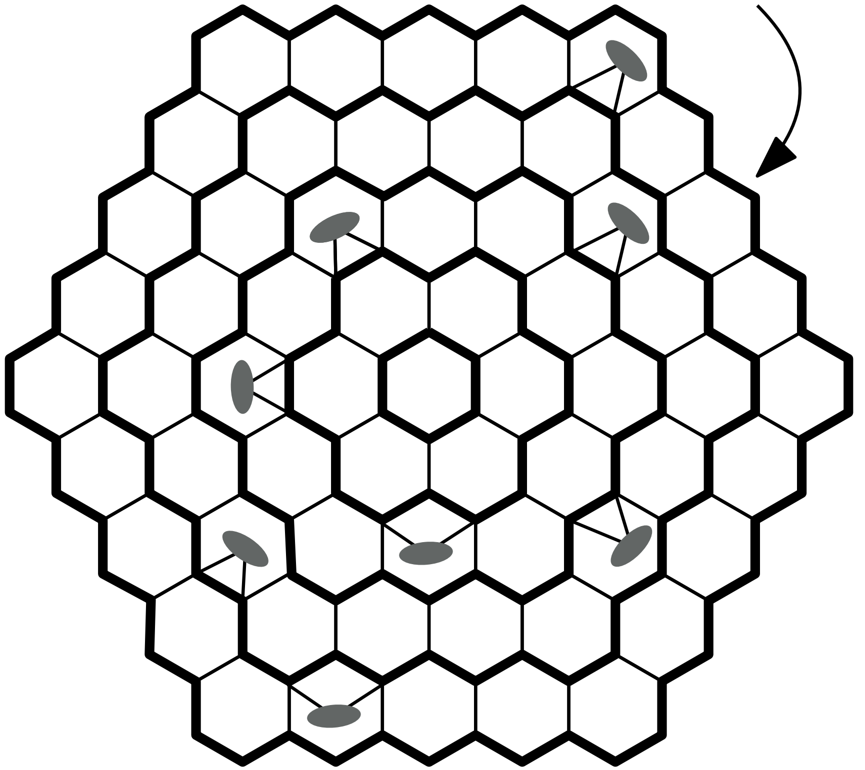
Bounding Treewidth



drawing of G with $\leq c$ crossings



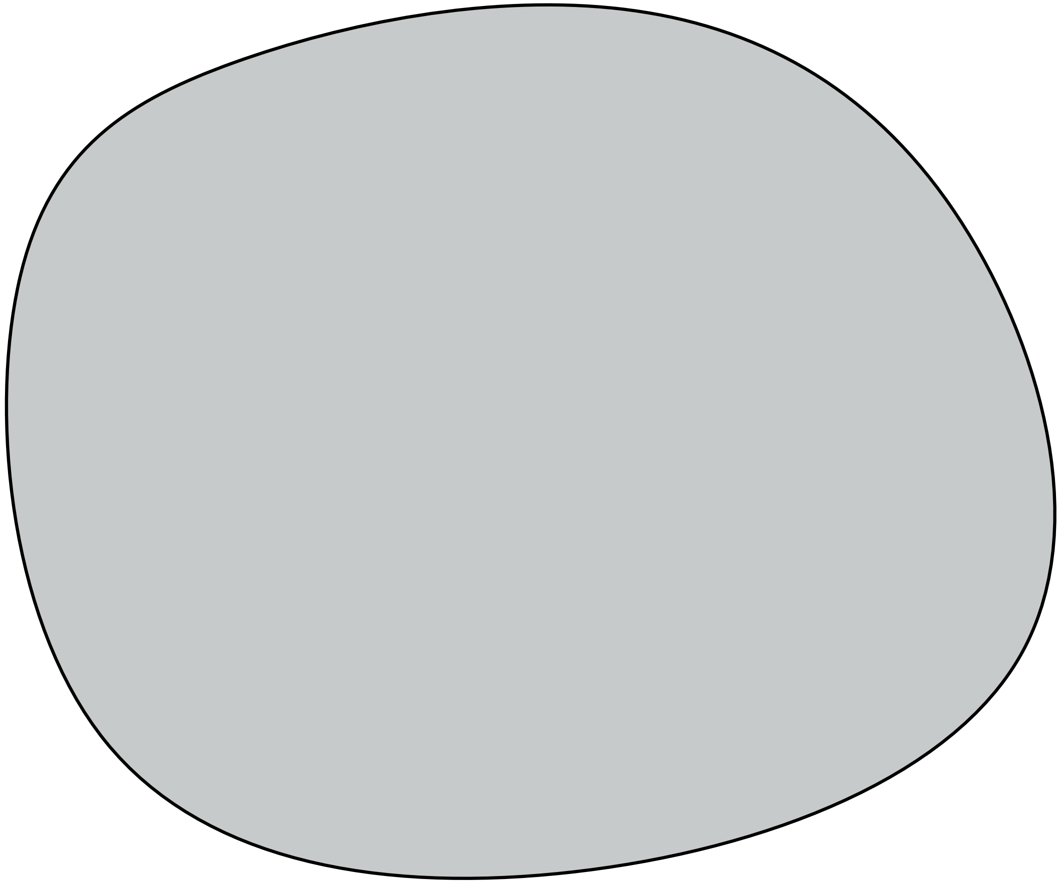
principal cycles

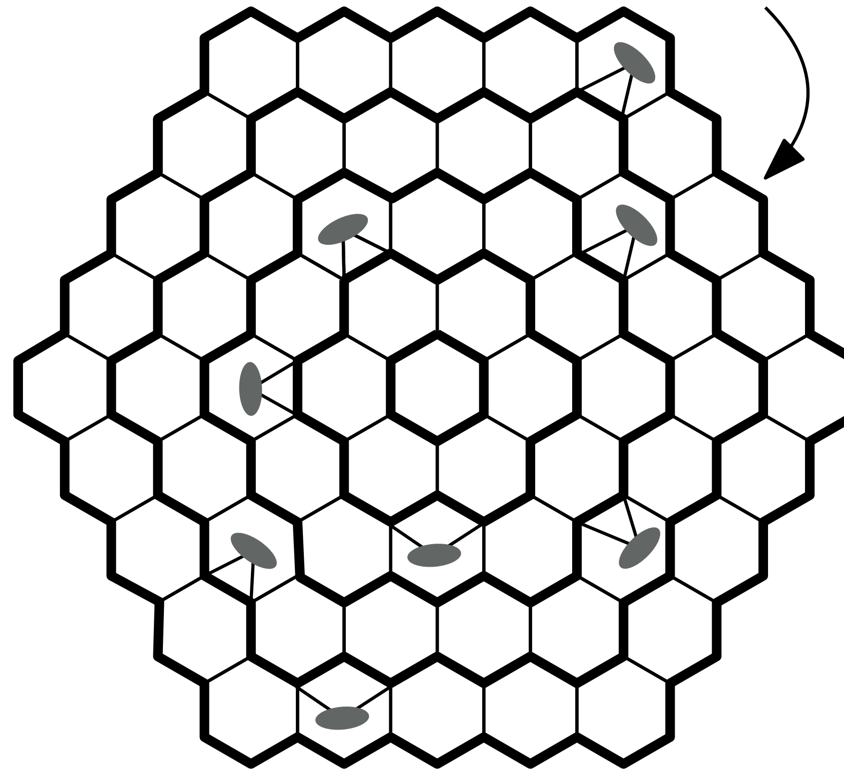


planar

Bounding Treewidth

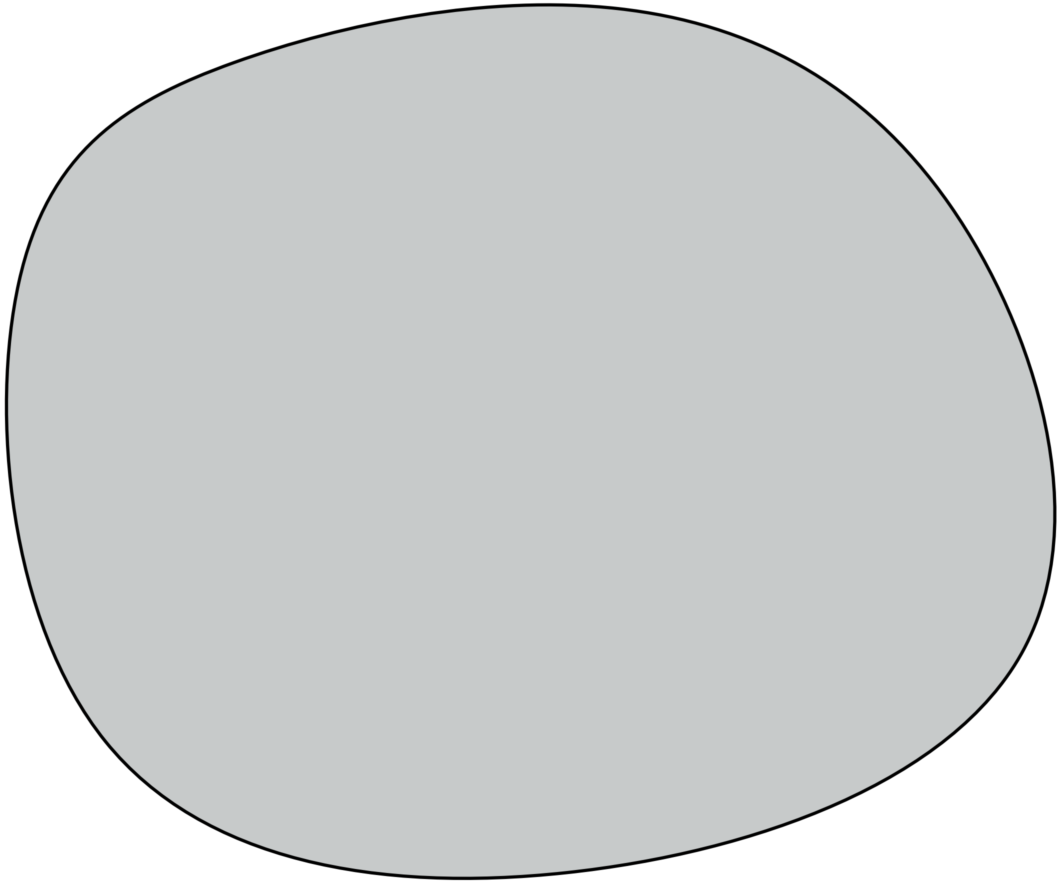
drawing of G with $\leq c$ crossings

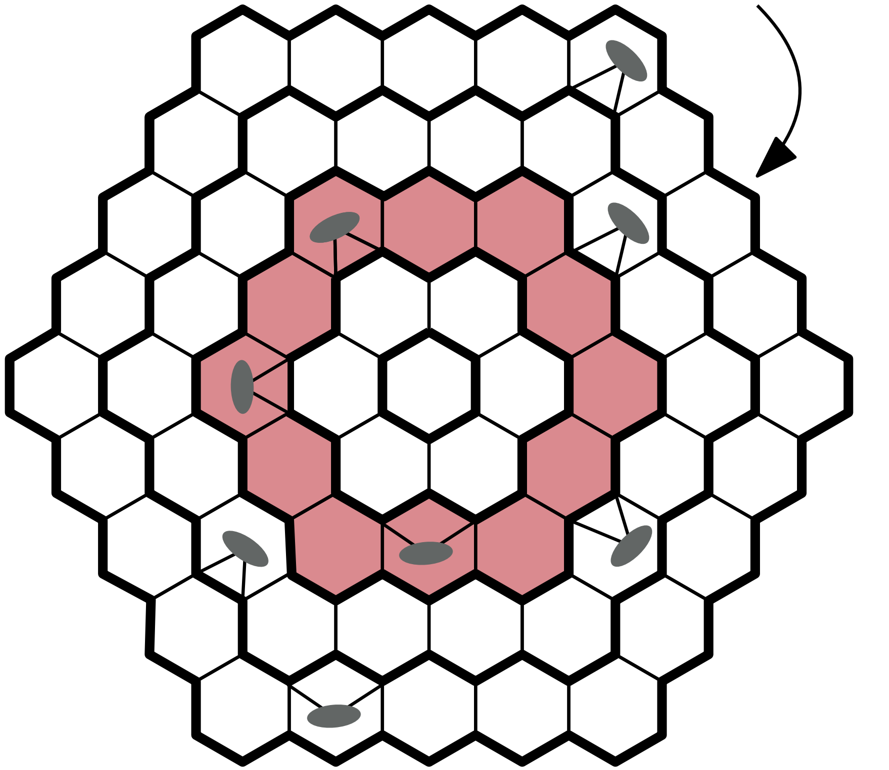


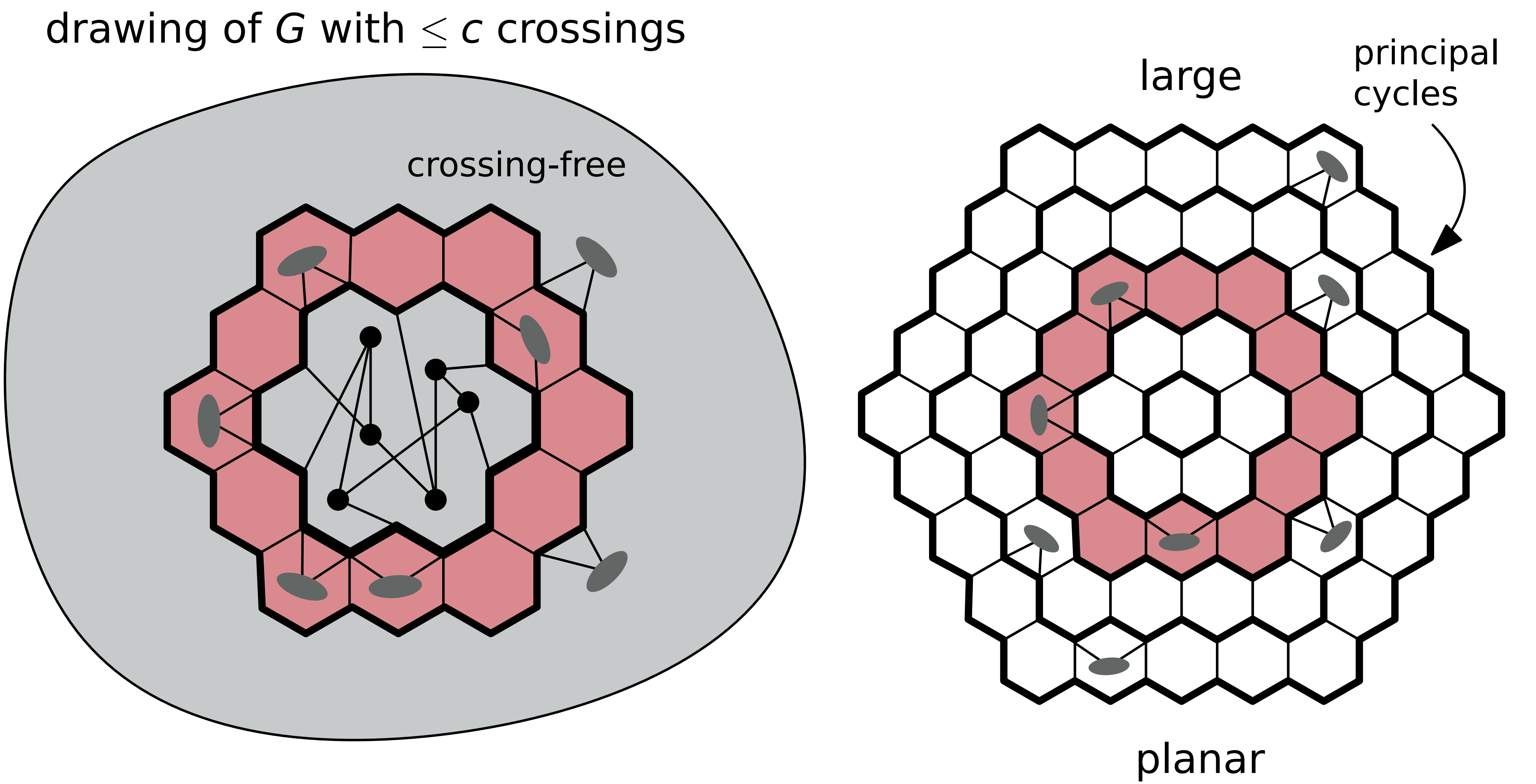


Bounding Treewidth

drawing of G with $\leq c$ crossings

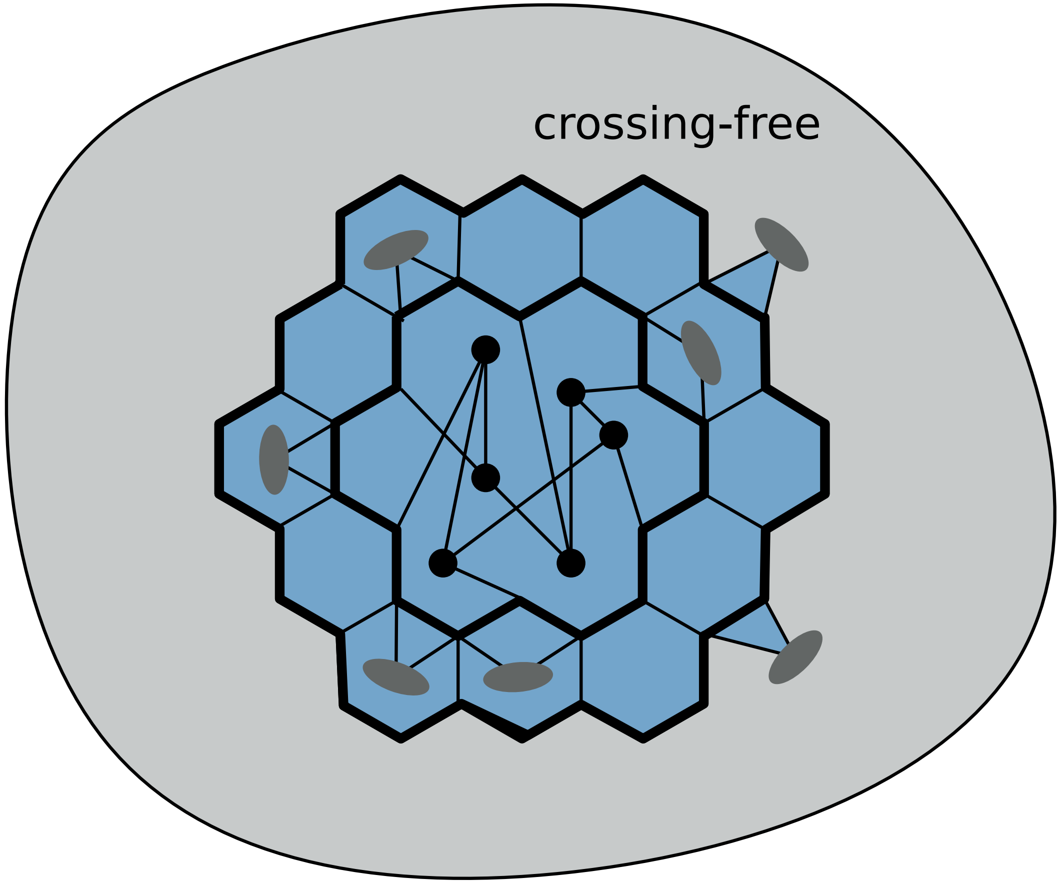


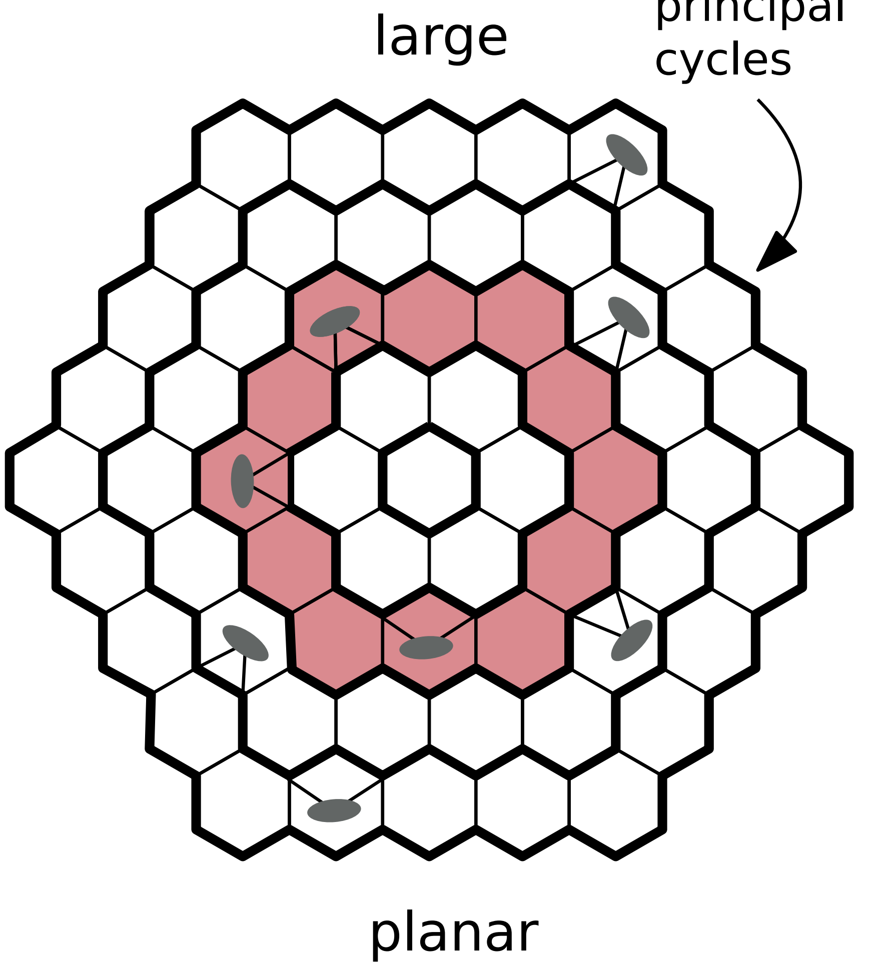



drawing of G with $\leq c$ crossings
crossing-free
large
principal cycles
planar

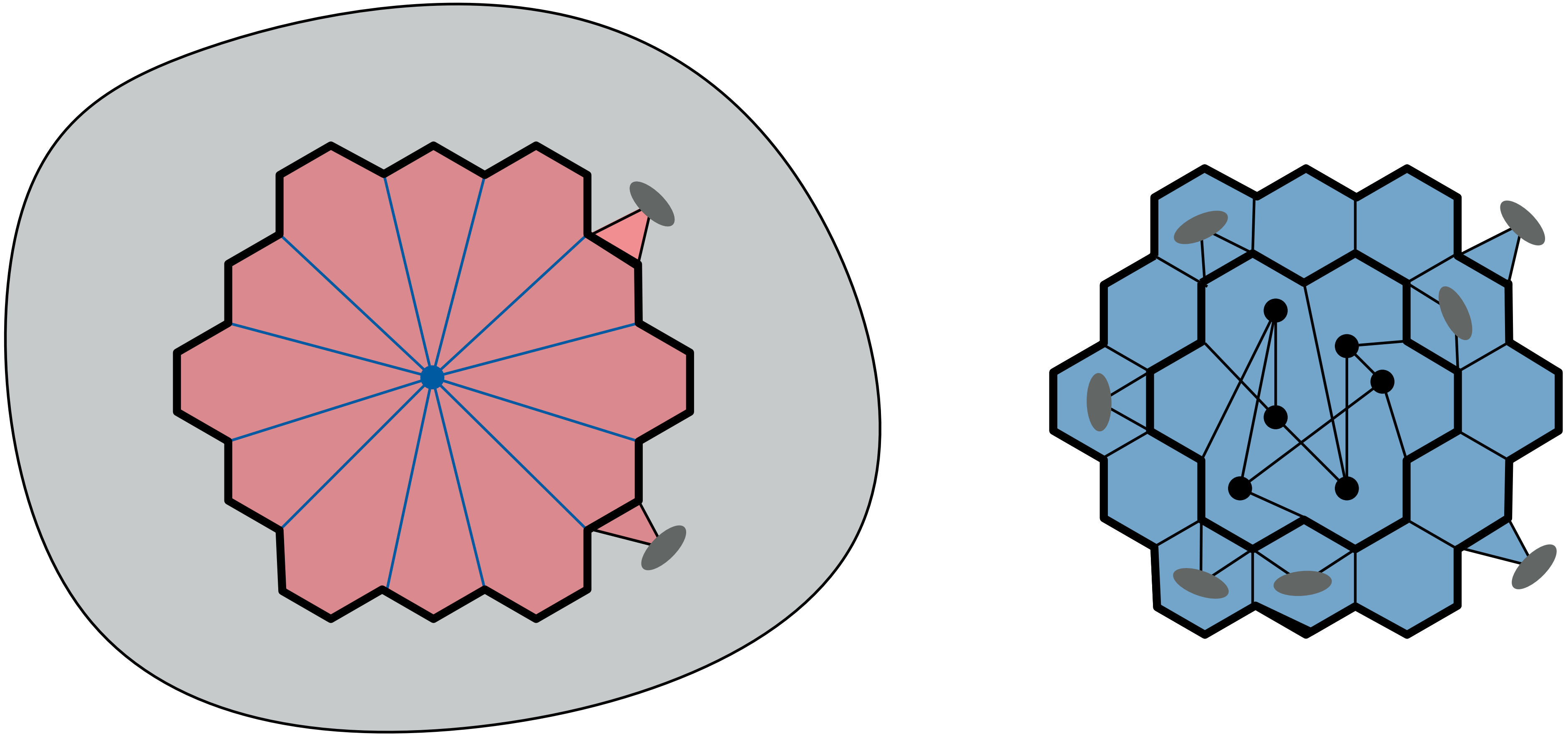
Bounding Treewidth

drawing of G with $\leq c$ crossings



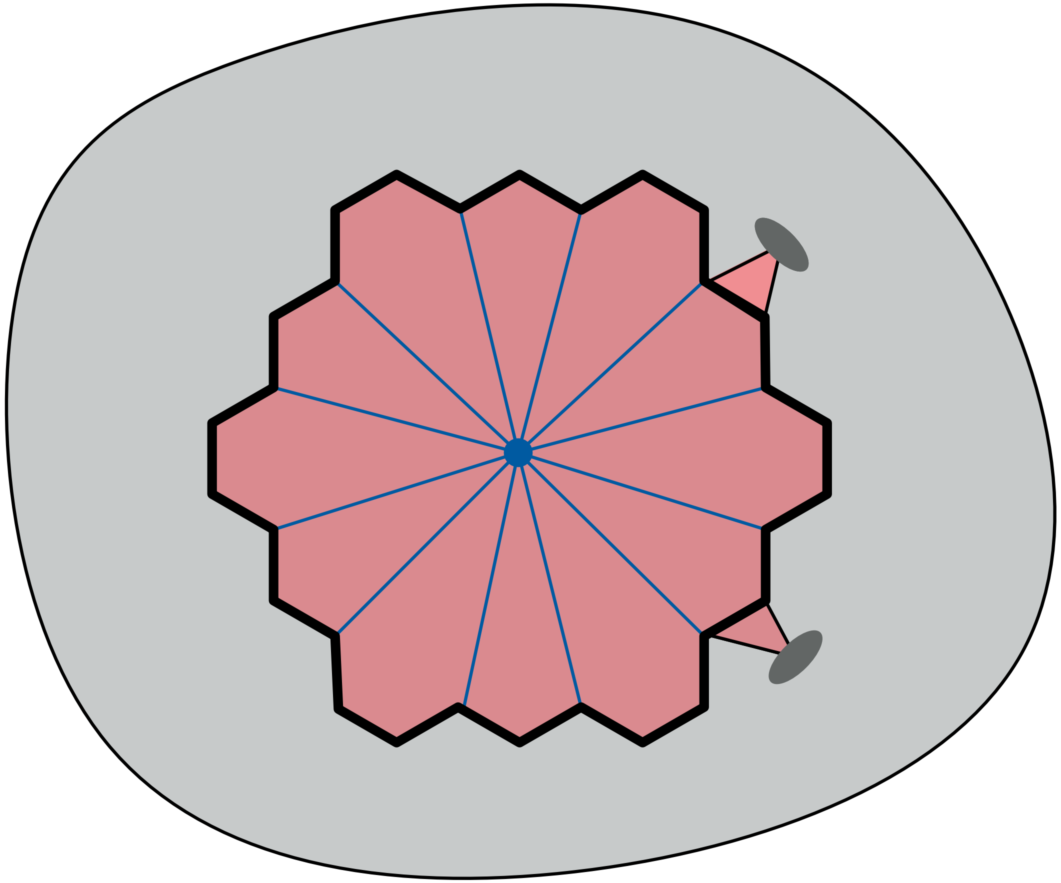


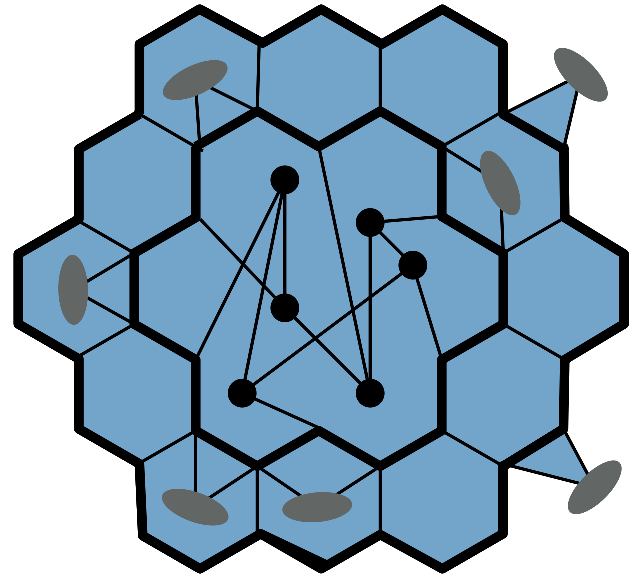
drawing of G' with $\leq c$ crossings



Bounding Treewidth

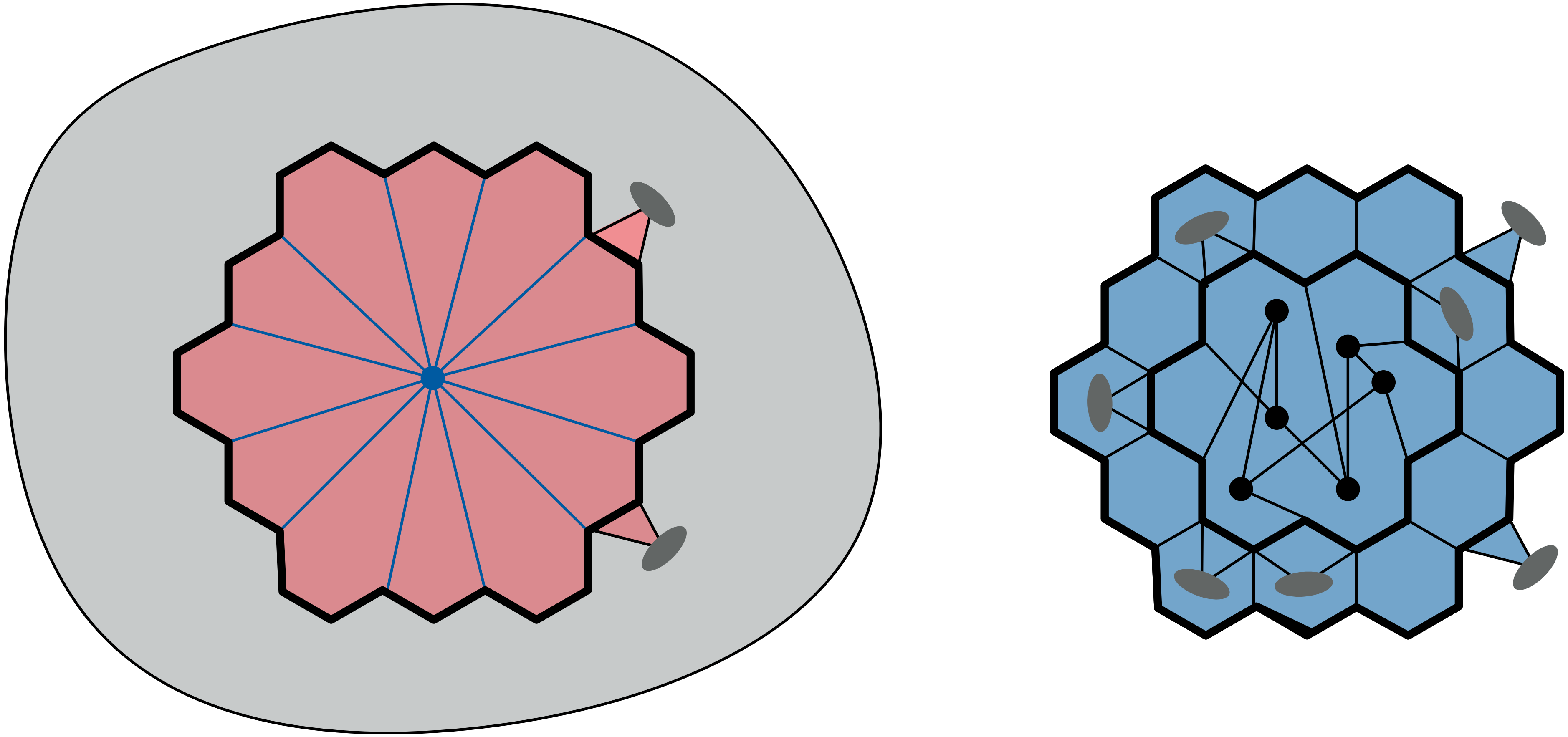
drawing of G' with $\leq c$ crossings





- $|V(G')| < |V(G)|$

drawing of G' with $\leq c$ crossings

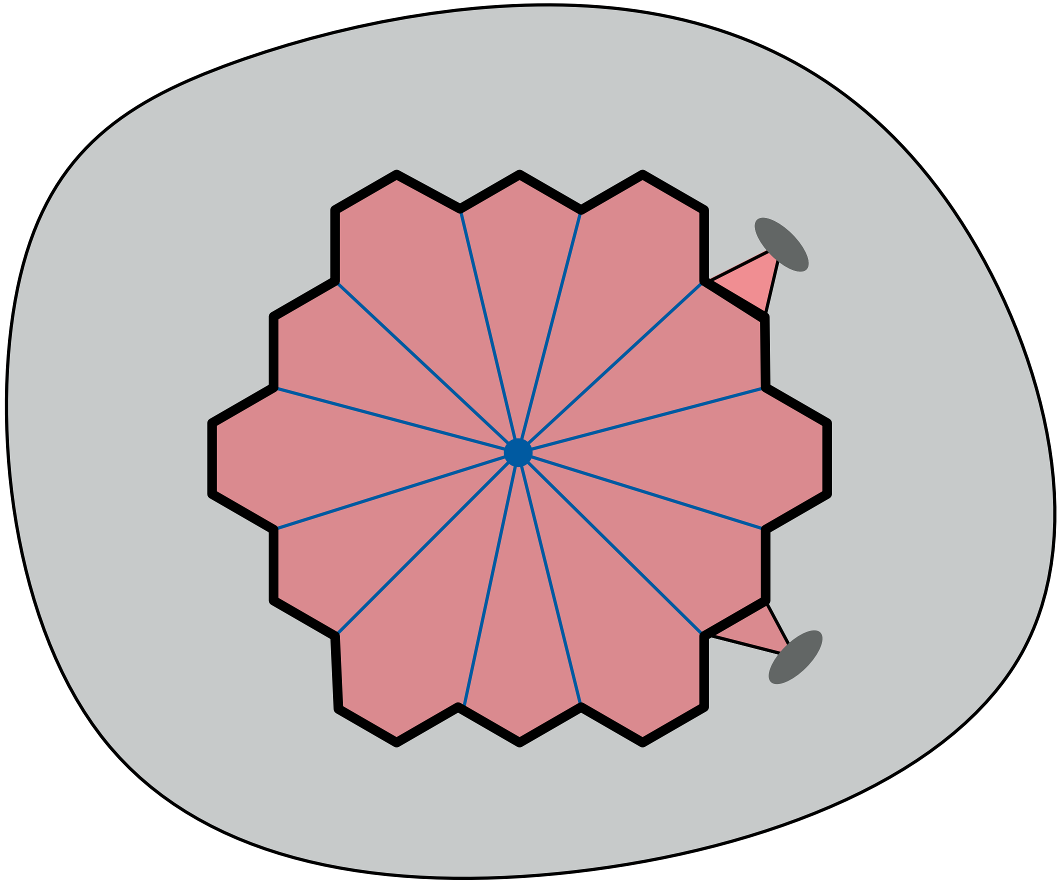


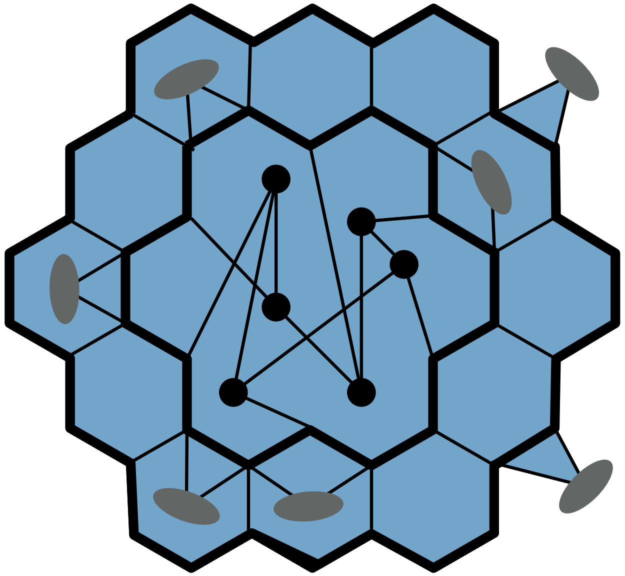
- $|V(G')| < |V(G)|$

G admits drawing with $\leq$ c crossings that avoids all patterns in $\mathcal{F}$

Bounding Treewidth

drawing of G' with $\leq c$ crossings



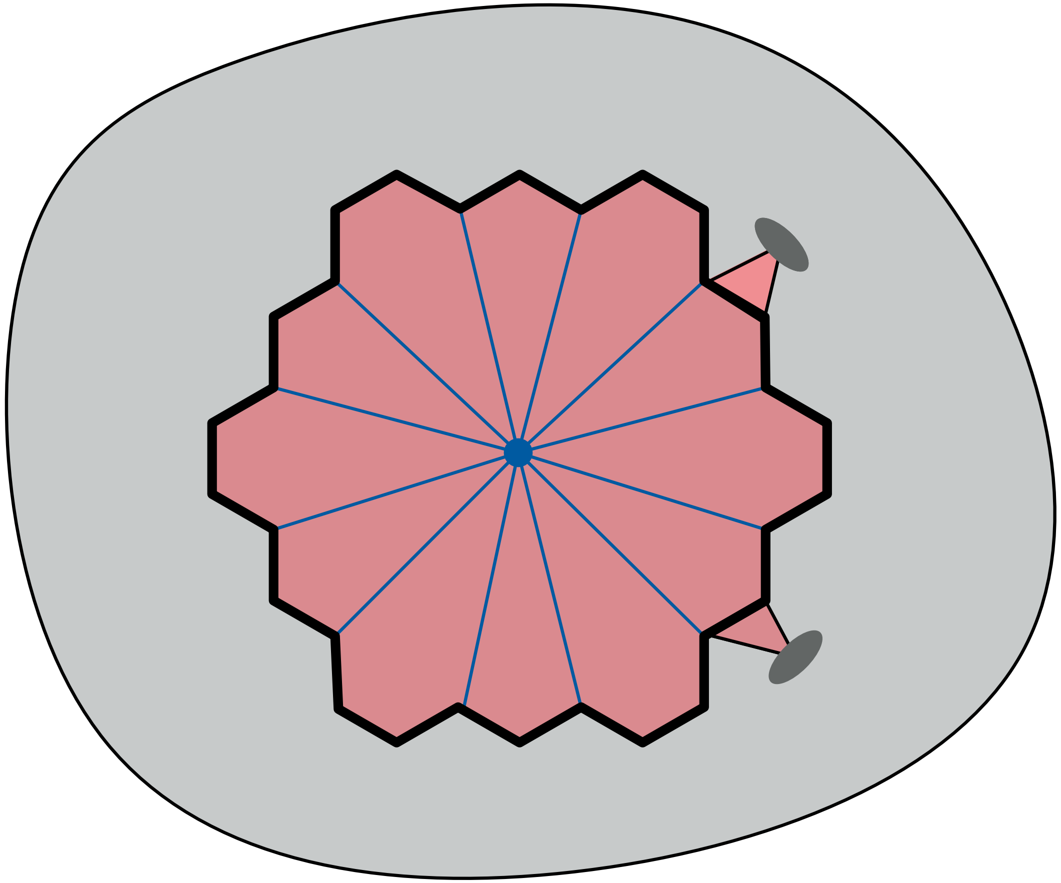


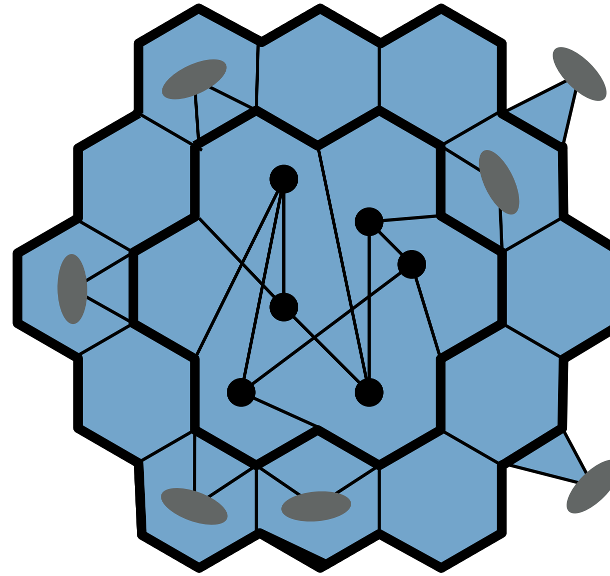
- $|V(G')| < |V(G)|$

G admits drawing with $\leq$ c crossings that avoids all patterns in $\mathcal{F}$

$\Rightarrow$ G' admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

drawing of G' with $\leq c$ crossings



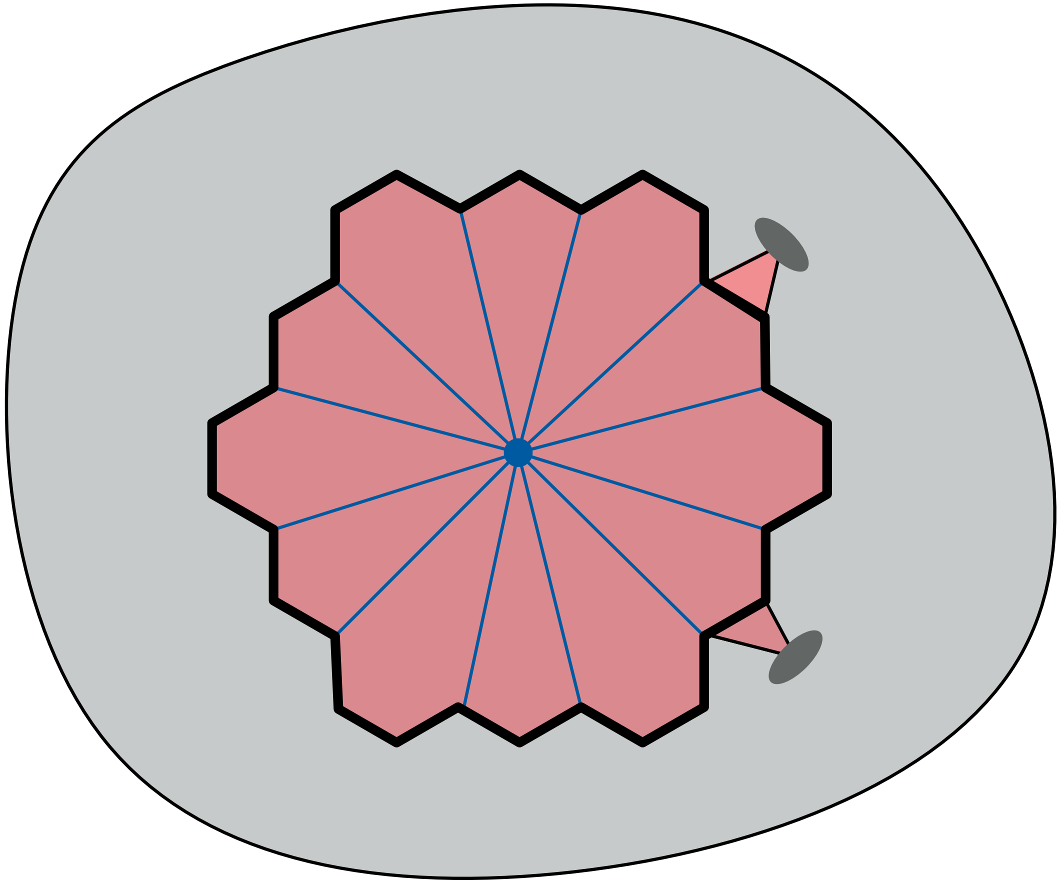


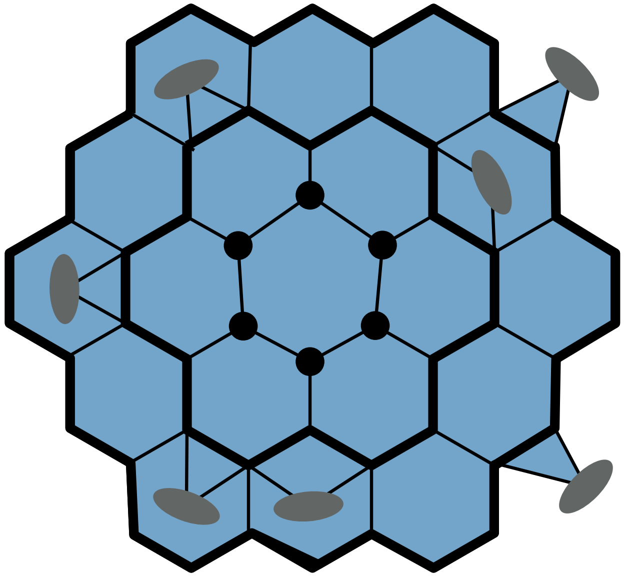
- $|V(G')| < |V(G)|$

  G' admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

Bounding Treewidth

drawing of G' with $\leq c$ crossings





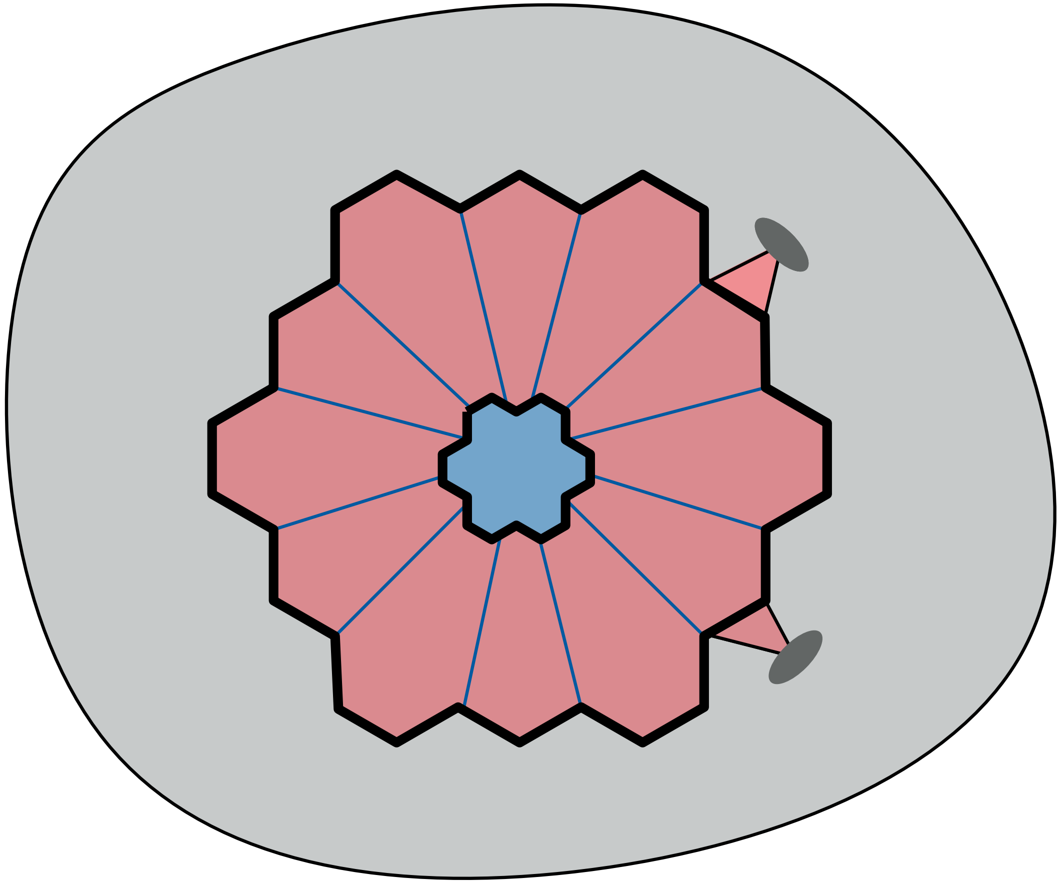
planar

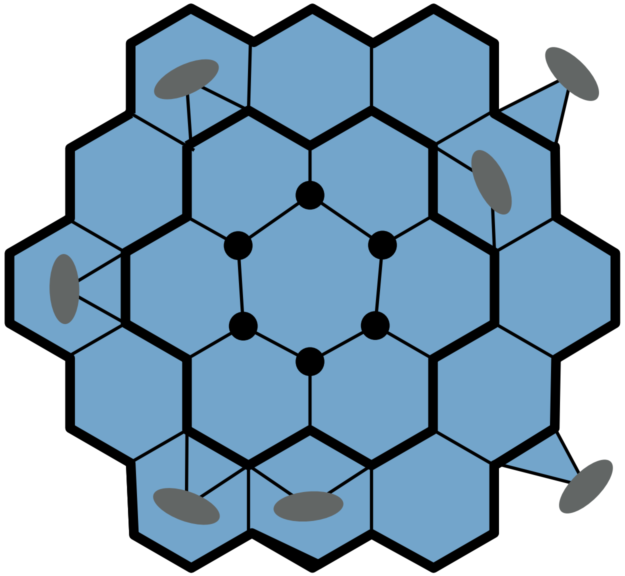
- $|V(G')| < |V(G)|$

G' admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

Bounding Treewidth

drawing of G' with $\leq c$ crossings



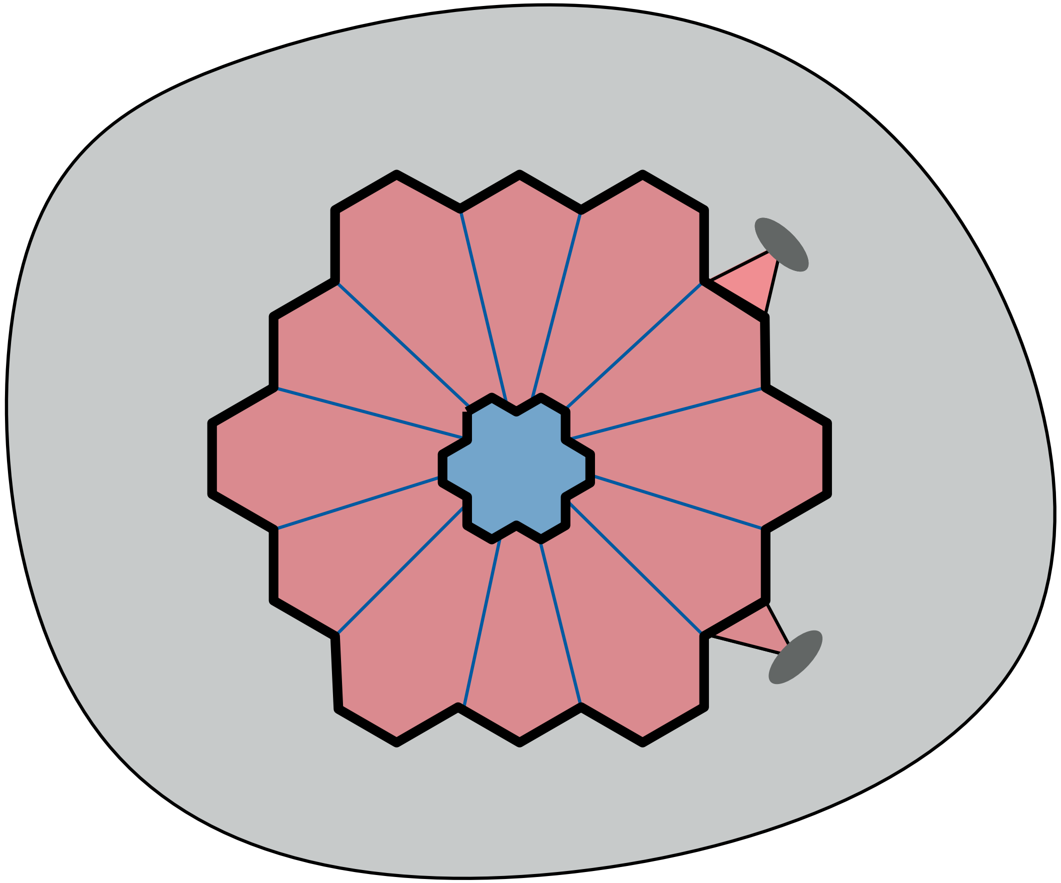


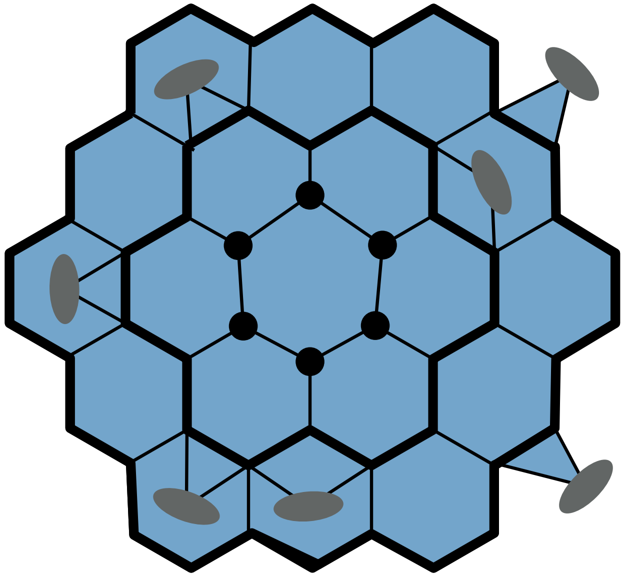
- $|V(G')| < |V(G)|$

  G' admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

Bounding Treewidth

drawing of G' with $\leq c$ crossings





planar

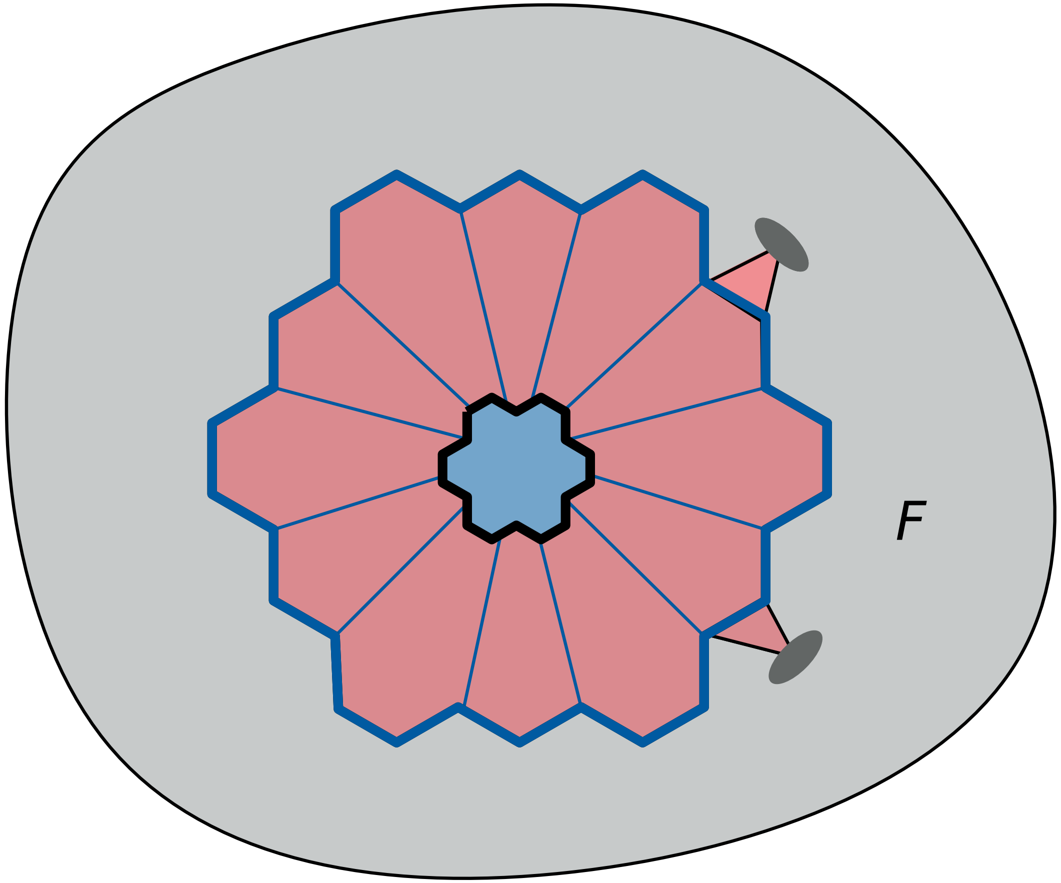
- $|V(G')| < |V(G)|$

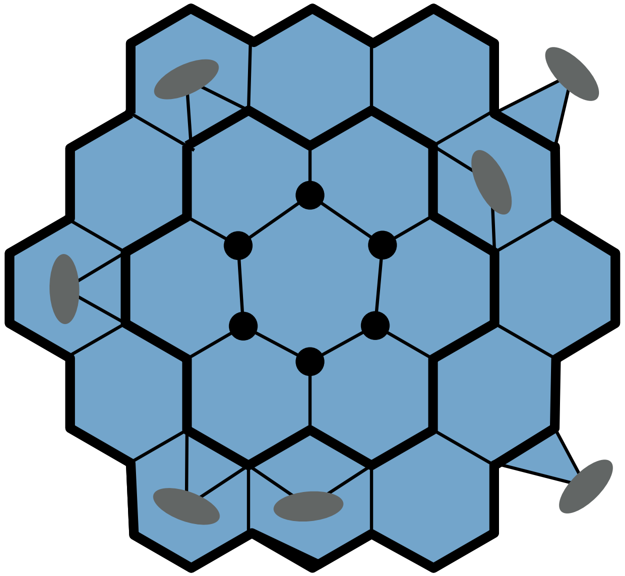
G' admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

$\Rightarrow$ G admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

Bounding Treewidth

drawing of G' with $\leq c$ crossings





- $|V(G')| < |V(G)|$

all blue edges uncrossed!

G' admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

$\Rightarrow$ G admits a drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$

Bounding Treewidth

Goal:

For any fixed set $\mathcal{F}$ of crossing patterns, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Lemma [Grohe]

There is a linear-time algo that, given a graph G, either

- recognizes: every $\mathcal{F}$-free drawing has $> c$ crossings, reject!
- recognizes $\mathrm{tw}(G) \leq w$ or move to Phase 2
- finds large hex-grid in G.

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \vDash G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

- no edge in $F \subseteq E$ is involved in a crossing
- no pattern in $\mathcal{F}$ contained
- $\leq c$ crossings

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \vDash G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

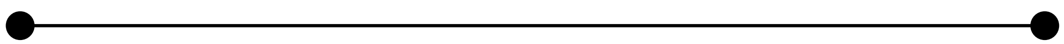
$\rightarrow$ express existence of suitable planarization

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \models G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization

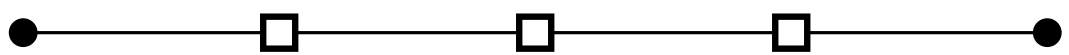
$e \in E \setminus F$



Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \models G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization

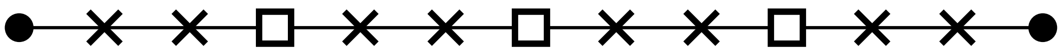


- c crossing dummies

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \vDash G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization

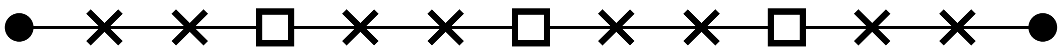


- c crossing dummies
- 2 subdivision dummies

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \models G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization



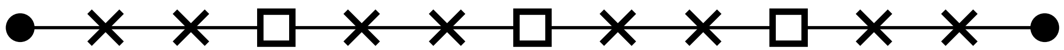
- c crossing dummies
- 2 subdivision dummies

Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \vDash G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization



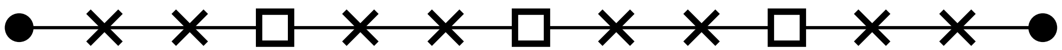
- c crossing dummies
- 2 subdivision dummies

Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$ vertices in C, x_i identified with y_i

Formulation in MSO_2

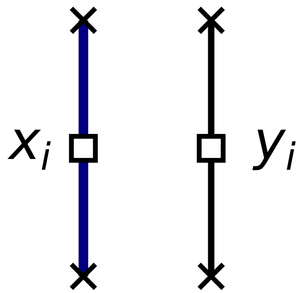
Goal: $\varphi_{\mathcal{F}}(F) \models G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization



- c crossing dummies
- 2 subdivision dummies

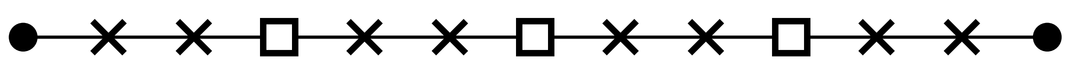
Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$ vertices in C, x_i identified with y_i



Formulation in MSO_2

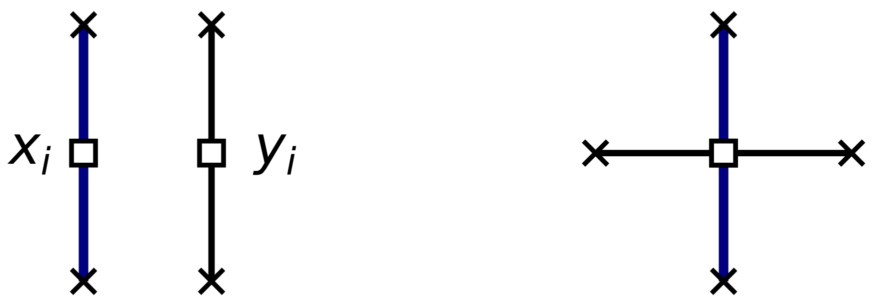
Goal: $\varphi_{\mathcal{F}}(F) \vDash G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization



- c crossing dummies
- 2 subdivision dummies

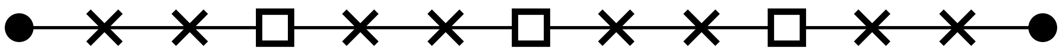
Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$ vertices in C, x_i identified with y_i



Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \models G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization



- c crossing dummies
- 2 subdivision dummies

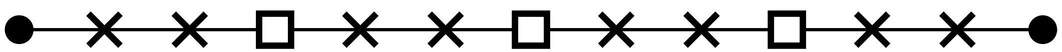
Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$ vertices in C, x_i identified with y_i

- pairwise distinct (except $x_i = y_i$)

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \vDash G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization



- c crossing dummies
- 2 subdivision dummies

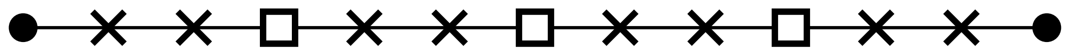
Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$ vertices in C, x_i identified with y_i

- pairwise distinct (except $x_i = y_i$)
- no self-crossings

Formulation in MSO_2

Goal: $\varphi_{\mathcal{F}}(F) \models G \Leftrightarrow$ G has $(F, \mathcal{F}, c)$-good drawing

$\rightarrow$ express existence of suitable planarization

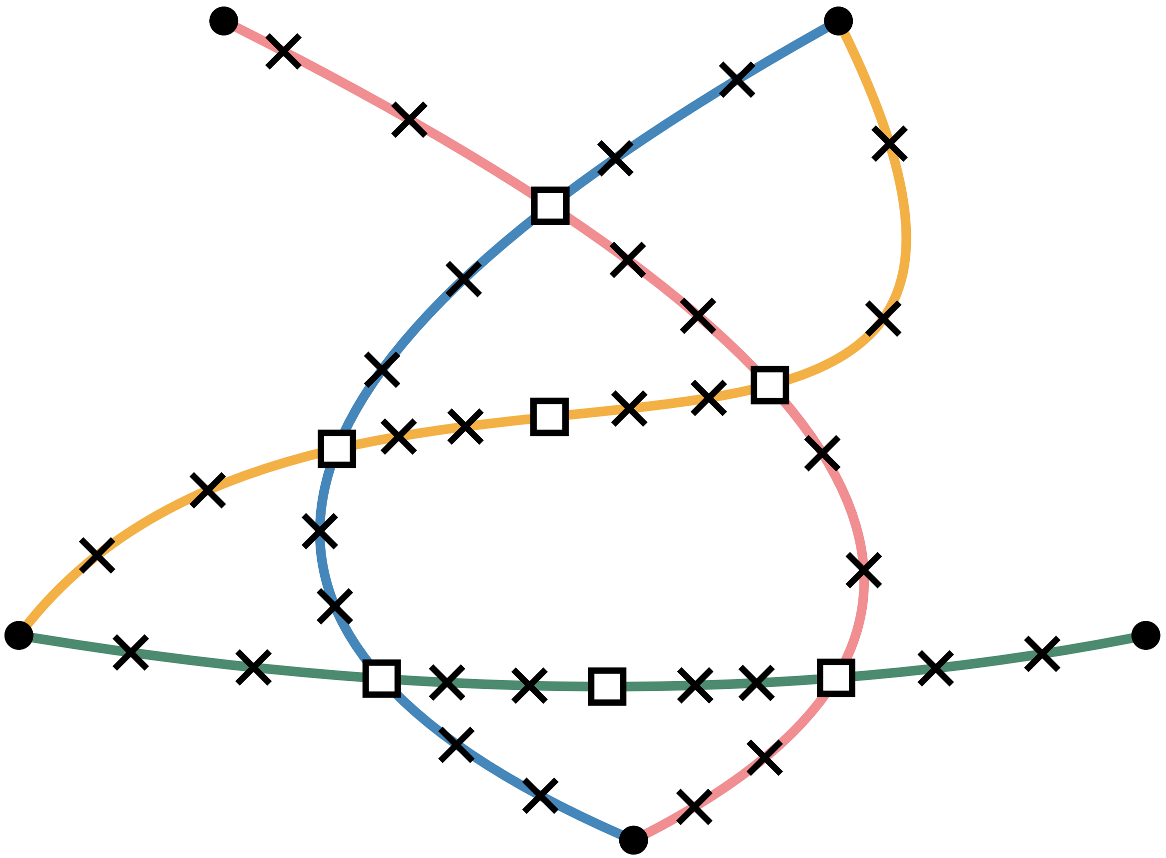


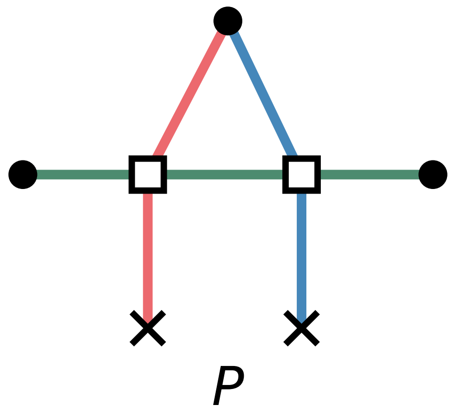
- c crossing dummies
- 2 subdivision dummies

Input: free variables $x_1, \ldots x_c$, $y_1, \ldots, y_c$ vertices in C, x_i identified with y_i

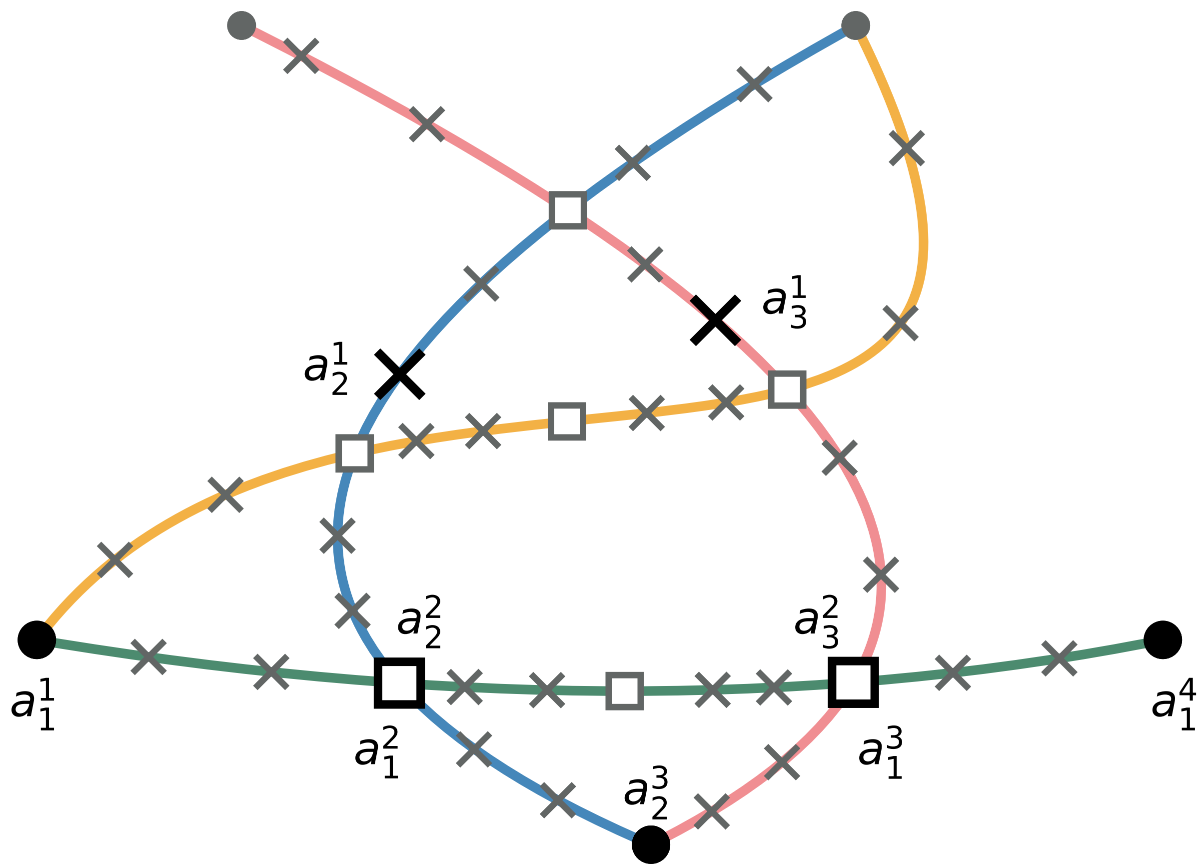
- pairwise distinct (except $x_i = y_i$)
- no self-crossings
- described graph is planar

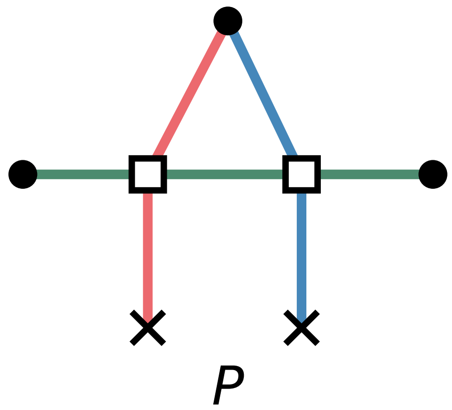
Forbid Single Pattern





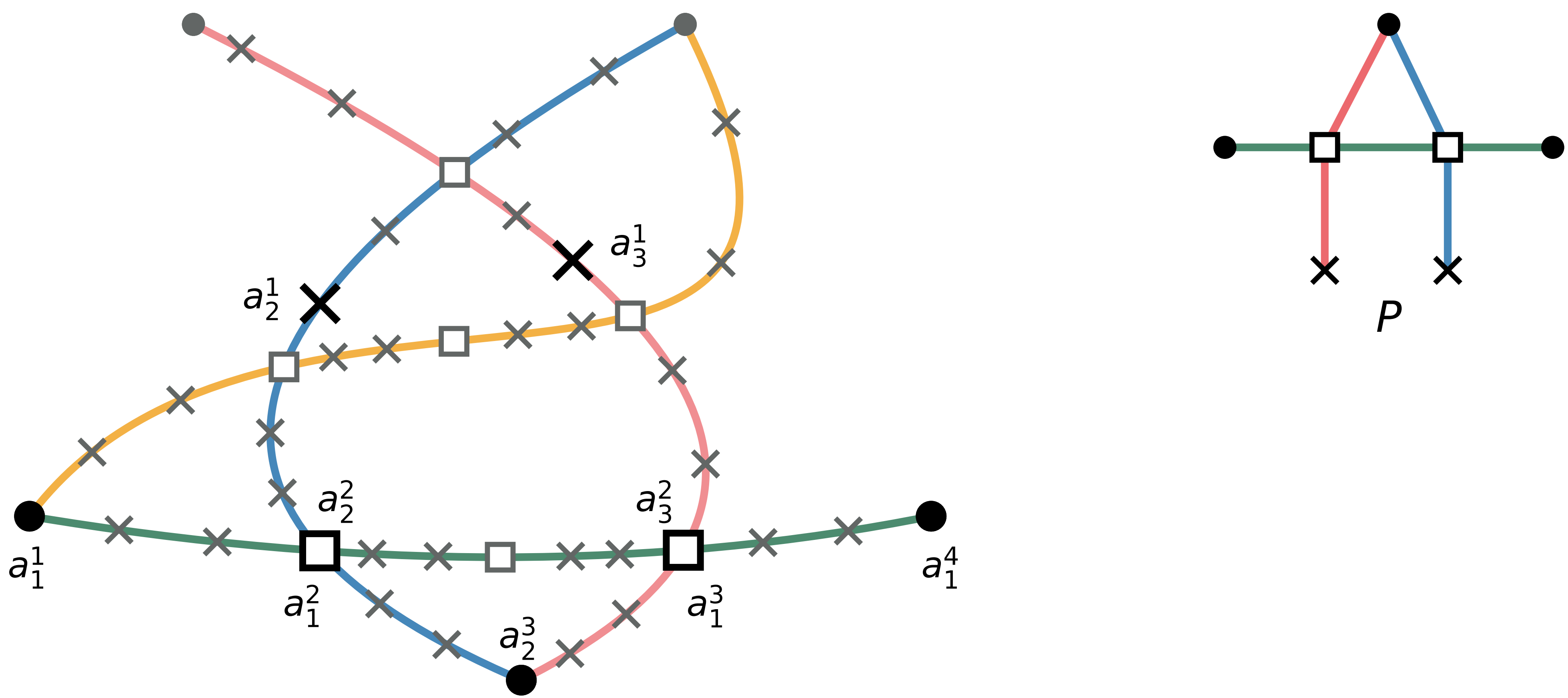
Forbid Single Pattern





$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

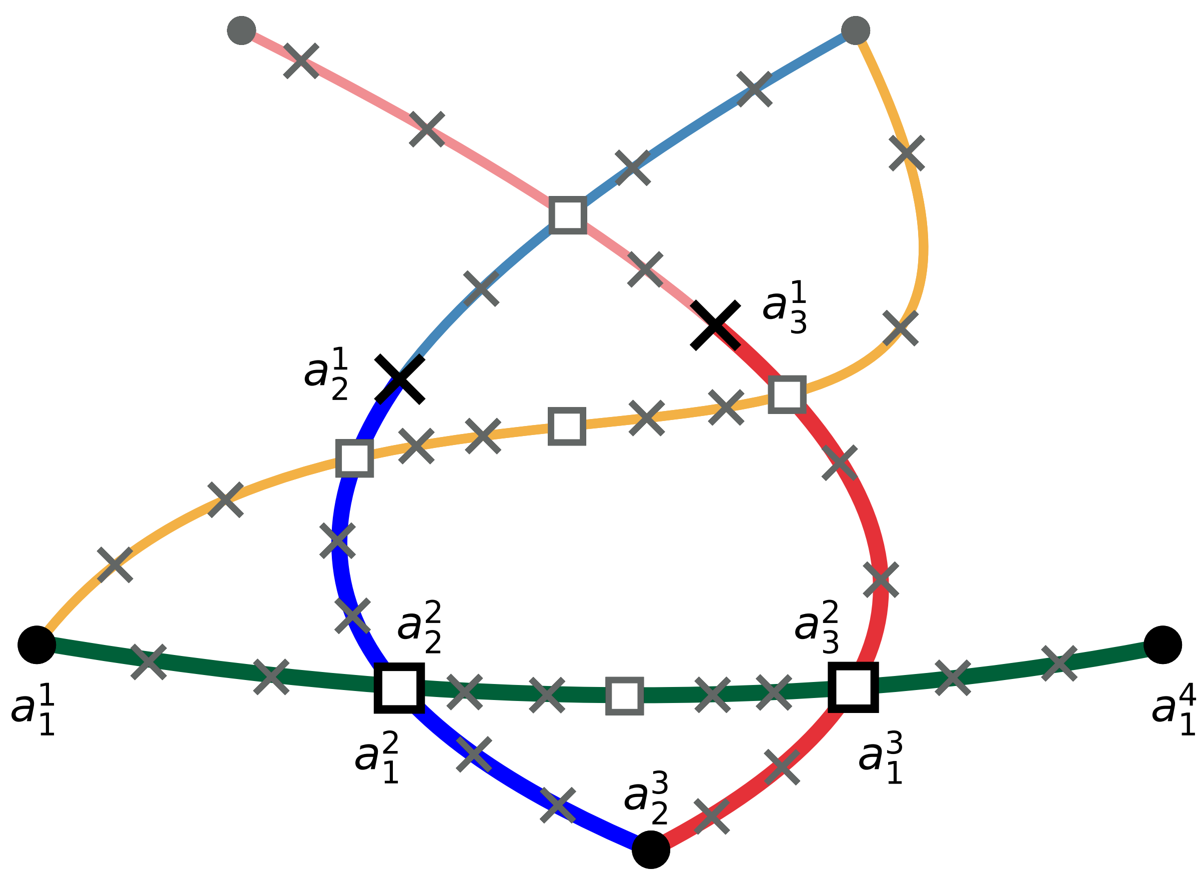
Forbid Single Pattern

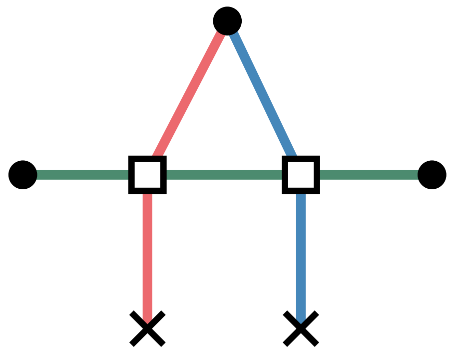


$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

Forbid Single Pattern



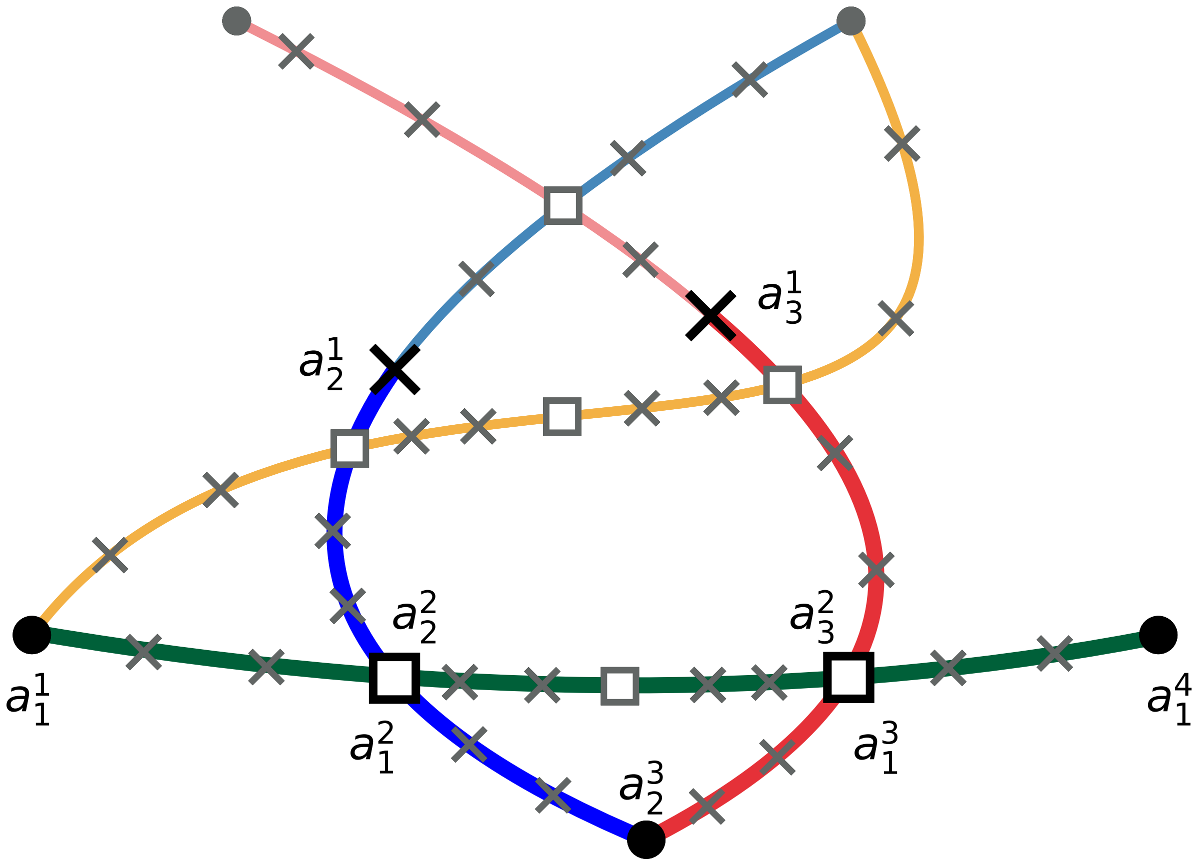


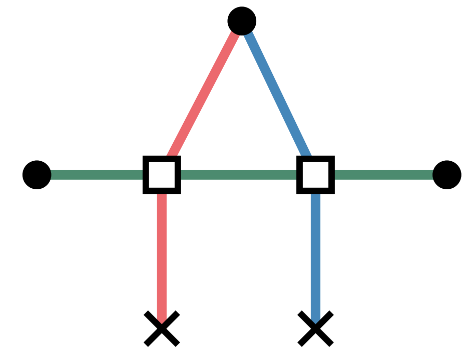
$$\varphi_P = \exists a_1^1 \in R \ \ \exists a_1^2, a_1^3 \in C \ \ \exists a_1^4 \in R \ \ \exists a_2^1, a_3^1 \in S \ \ \exists a_2^2, a_3^2 \in C \ \ \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

$$\exists E_1, E_2, E_3 \subseteq E$$

Forbid Single Pattern





P

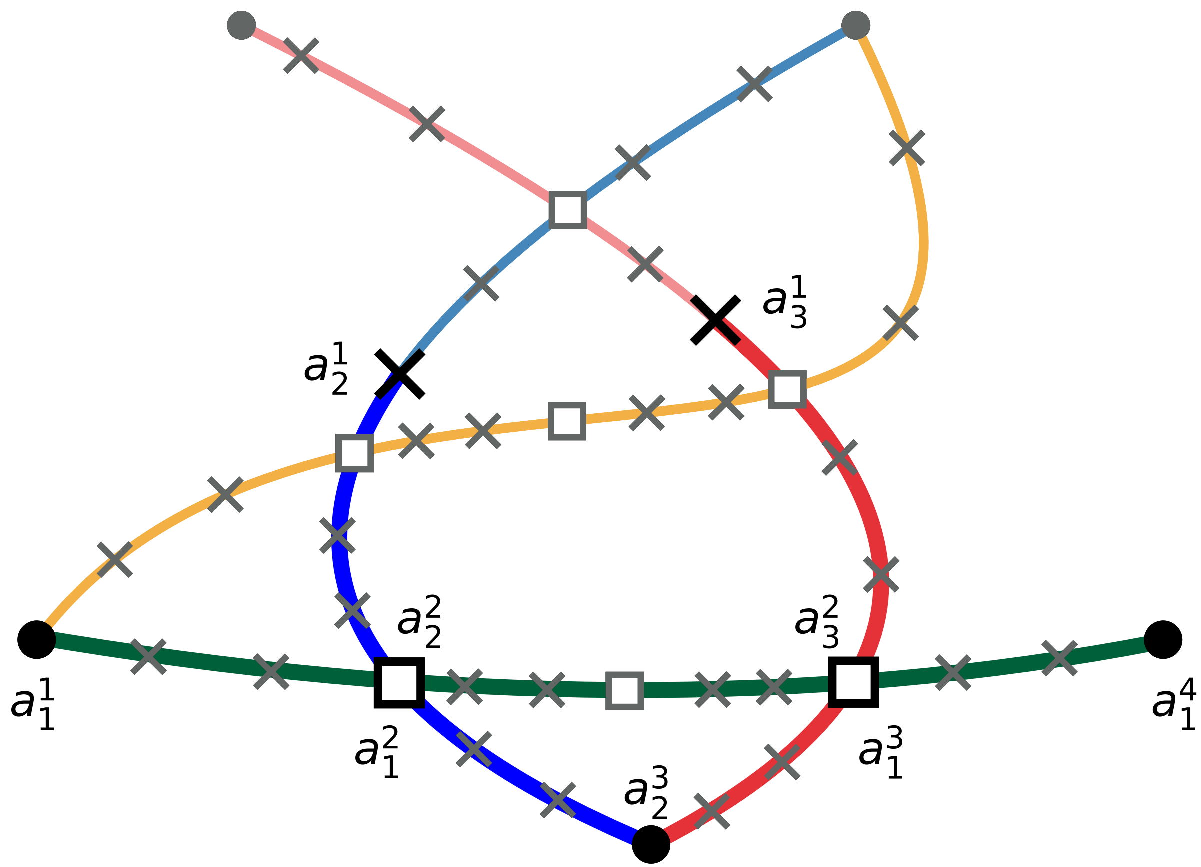
$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

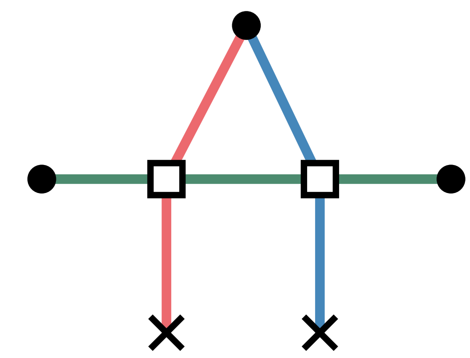
$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

$$\exists E_1, E_2, E_3 \subseteq E$$

$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

Forbid Single Pattern





$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

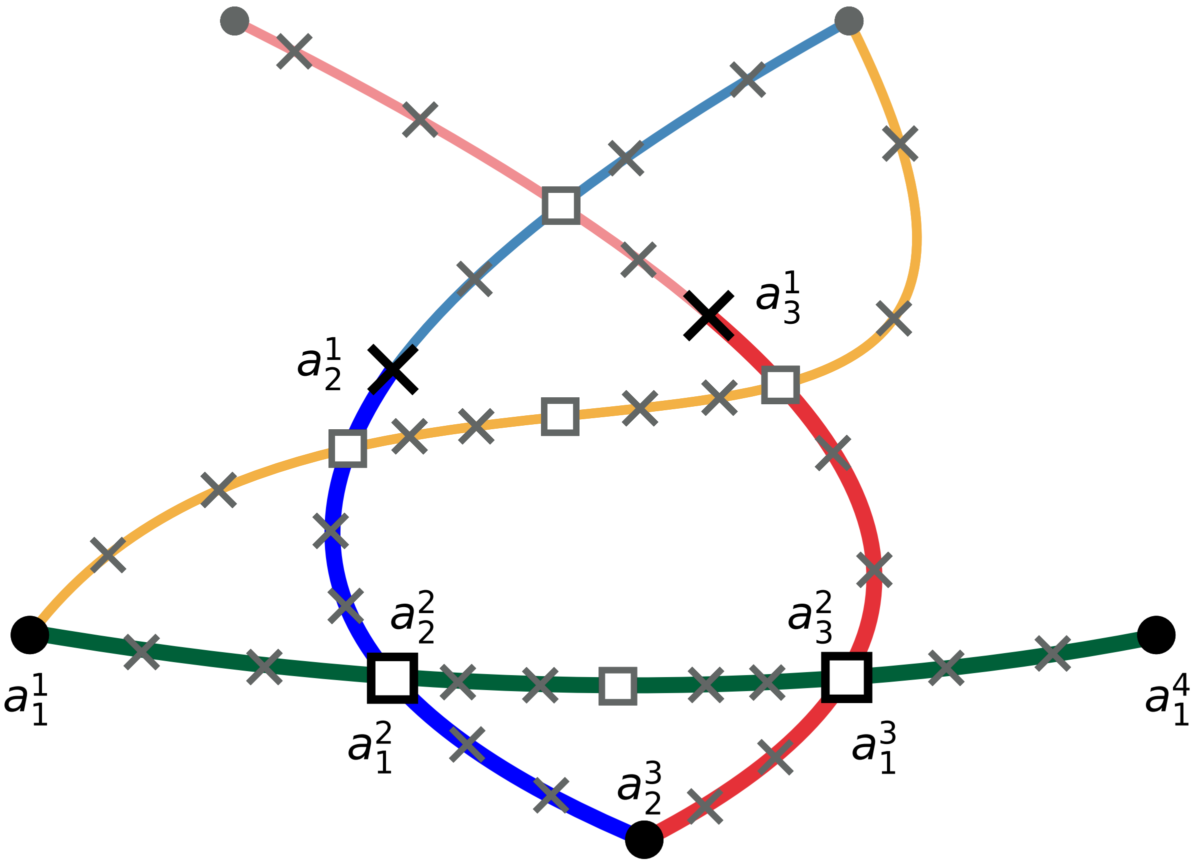
$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

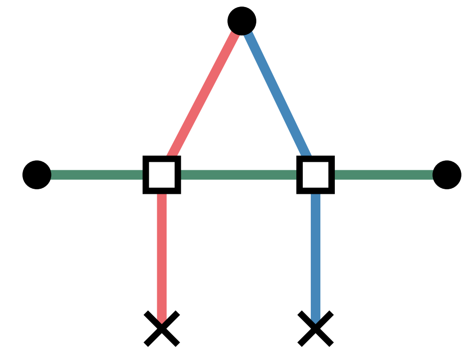
$$\exists E_1, E_2, E_3 \subseteq E$$

$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \ldots) \wedge \text{chain}(E_3, \ldots)$$

Forbid Single Pattern





P

chain = path s.t.

- internal vertices in $C \cup S$
- given vertices in correct order

$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

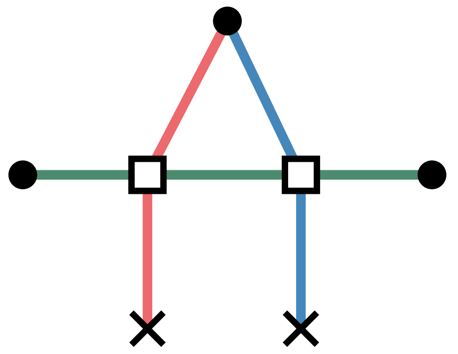
$$\exists E_1, E_2, E_3 \subseteq E$$

$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \ldots) \wedge \text{chain}(E_3, \ldots)$$

Forbid Single Pattern

$\varphi_{\mathcal{F}} = \text{planar}^{\times} \wedge \bigwedge_{P \in \mathcal{F}} \neg \varphi_P$



chain = path s.t.

- internal vertices in $C \cup S$
- given vertices in correct order

$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

$$\exists E_1, E_2, E_3 \subseteq E$$

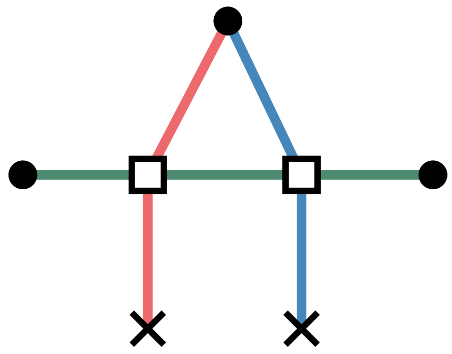
$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \ldots) \wedge \text{chain}(E_3, \ldots)$$

Forbid Single Pattern

$\varphi_{\mathcal{F}} = \text{planar}^{\times} \wedge \bigwedge_{P \in \mathcal{F}} \neg \varphi_P$

- size depends only on c and pattern-size



chain = path s.t.

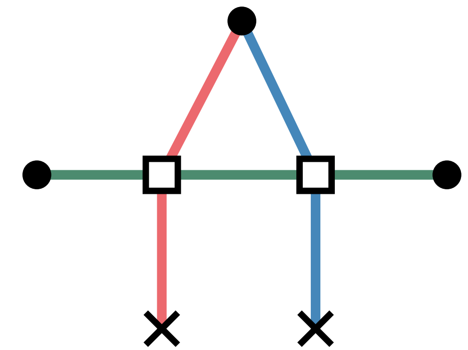
- internal vertices in $C \cup S$
- given vertices in correct order

$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$
$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$
$$\exists E_1, E_2, E_3 \subseteq E$$
$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$
$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \ldots) \wedge \text{chain}(E_3, \ldots)$$

Forbid Single Pattern

$\varphi_{\mathcal{F}} = \text{planar}^{\times} \wedge \bigwedge_{P \in \mathcal{F}} \neg \varphi_P$

- size depends only on c and pattern-size
- treewidth does not increase by subdividing edges



chain = path s.t.

- internal vertices in $C \cup S$
- given vertices in correct order

$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

$$\exists E_1, E_2, E_3 \subseteq E$$

$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

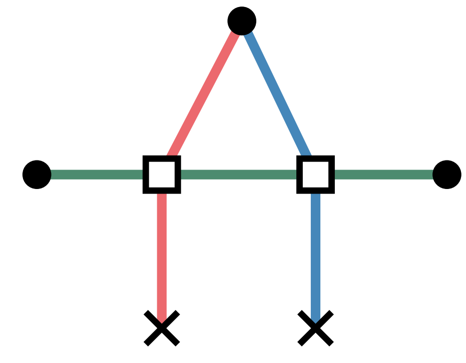
$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \dots) \wedge \text{chain}(E_3, \dots)$$

Forbid Single Pattern

$\varphi_{\mathcal{F}} = \text{planar}^{\times} \wedge \bigwedge_{P \in \mathcal{F}} \neg \varphi_P$

- size depends only on c and pattern-size
- treewidth does not increase by subdividing edges

↓ Courcelle



chain = path s.t.

- internal vertices in $C \cup S$
- given vertices in correct order

$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

$$\exists E_1, E_2, E_3 \subseteq E$$

$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

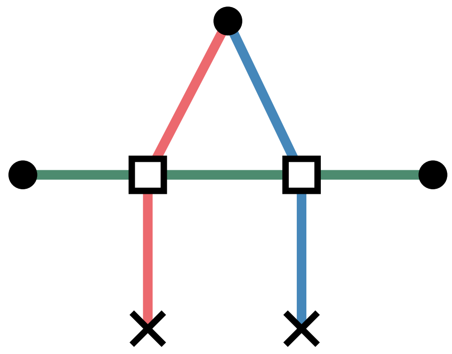
$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \ldots) \wedge \text{chain}(E_3, \ldots)$$

Forbid Single Pattern

$\varphi_{\mathcal{F}} = \text{planar}^{\times} \wedge \bigwedge_{P \in \mathcal{F}} \neg \varphi_P$

- size depends only on c and pattern-size
- treewidth does not increase by subdividing edges

Courcelle



Theorem
For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with $\leq c$ crossings that avoids all patterns in $\mathcal{F}$ is FPT w.r.t. c.

chain = path s.t.

- internal vertices in $C \cup S$
- given vertices in correct order

$$\varphi_P = \exists a_1^1 \in R \;\; \exists a_1^2, a_1^3 \in C \;\; \exists a_1^4 \in R \;\; \exists a_2^1, a_3^1 \in S \;\; \exists a_2^2, a_3^2 \in C \;\; \exists a_2^3 \in R$$

$$\text{isCrossing}(a_1^2, a_2^2) \wedge \text{isCrossing}(a_1^3, a_3^2)$$

$$\exists E_1, E_2, E_3 \subseteq E$$

$$\bigwedge_{i \neq j} \text{disjoint}(E_i, E_j)$$

$$\text{chain}(E_1, a_1^1, a_1^2, a_1^3, a_1^4) \wedge \text{chain}(E_2, \ldots) \wedge \text{chain}(E_3, \ldots)$$

Summary and Outlook

- combinatorial formalization crossing patterns

Summary and Outlook

- combinatorial formalization crossing patterns

Meta-Theorem

For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

Summary and Outlook

- combinatorial formalization crossing patterns

Meta-Theorem

For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

- extendable to 2-layer setting

Summary and Outlook

- combinatorial formalization crossing patterns

Meta-Theorem

For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

- extendable to 2-layer setting
- extendable to SEFE

Summary and Outlook

- combinatorial formalization crossing patterns

Meta-Theorem

For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

- extendable to 2-layer setting
- extendable to SEFE
- not applicable if graph class relies on topological properties

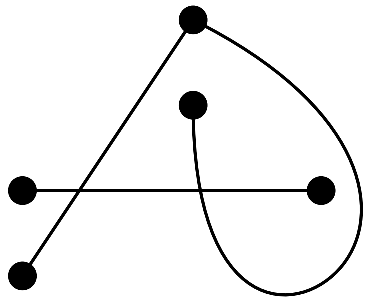
Summary and Outlook

- combinatorial formalization crossing patterns

Meta-Theorem

For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

- extendable to 2-layer setting
- extendable to SEFE
- not applicable if graph class relies on topological properties



Summary and Outlook

- combinatorial formalization crossing patterns

Meta-Theorem

For any fixed set of crossing patterns $\mathcal{F}$, testing whether a graph G admits drawing with at most c crossings that avoids all patterns in $\mathcal{F}$ is FPT with respect to c.

- extendable to 2-layer setting
- extendable to SEFE
- not applicable if graph class relies on topological properties